Reflections on the number field sieve

Cliffe, Emma Hazel

Award date:
2007

Awarding institution:
University of Bath

REFLECTIONS ON THE NUMBER FIELD SIEVE

Submitted by Emma Hazel Cliffe
for the degree of
Doctor of Philosophy
of the University of Bath
2007

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

Summary

The number field sieve is currently the asymptotically fastest factoring algorithm and is fastest in practice for integers greater than approximately 110 decimal digits.

We develop one of the existing estimation techniques for the quantity of data produced by the classical sieve, we draw attention to a possible underlying cause for the severe underestimate that has previously been recorded. We suggest a method for improving the estimates and give supportive evidence that reasonable estimates can be produced in this way.

We consider the special cases, where good parameters for the number field sieve were selected by hand, in the context of innovations made in polynomial selection in the general case. We note characteristics that are shared by a variety of special cases including the possibility of subfield structure.

We demonstrate how the general case polynomial selection methods are capable of isolating some sorts of special cases without guidance as to the structure of the number to be factored. We note that this blurs the previously sharp distinction between special cases, as defined by a set of unusual shared and convenient characteristics, and general cases. We pose some open questions regarding this and the possibility of repudiation or opportunistic attacks on the RSA cryptosystem.

Noting that this could raise the importance of advances in the algorithm that are applicable only in the special cases, we investigate the possibility of utilising the subfield structure. While there are promising facets of the natural method of utilising this structure it is shown not to be a practical method. We support this result using the estimation techniques mentioned above.

Acknowledgements

There are many people that have helped and supported me in my work. In particular I would like to mention James Davenport, my supervisor, for all his support and the many interesting and challenging discussions over the years. Andrew Holt, a fellow student, I must acknowledge for providing a sounding board throughout my studies and for the two and a half years of almost daily discussions regarding our work which were of an immense help to me. I would also like to thank John Fitch for keeping me going at the end and agreeing to sort out any scary forms that came his way.

I would like to thank all those people with whom I have shared an office for the interesting debates, proof reading, help with code debugging, skill at backgammon, intriguing ideas for wasting time and perhaps most importantly for all the cups of coffee.

I must acknowledge my parents Lorraine and Alan Jones and my brother Neil for their never ending patience regarding all efforts at communicating with me in the last 3 years and more generally for caring about my research because I do, without knowing that much about it.

My friends, I would like to thank for listening to my various ramblings about work, I can't mention you all but the most staunch allies have been Michael Stephenson, Richard Dzien, Jamie Stone and Jim Grimmett. Many apologies for boring you all!

Finally, I would like to thank Owen for all the love and support, I know I could not have finished this without you.

Contents

Chapter 1

Introduction

The integer factorisation problem is: given a positive integer n find the prime factorisation $n = p_1^{e_1} p_2^{e_2} \dots p_k^{e_k}$ where the p_i, $i = 1, \dots, k$ are pairwise distinct primes and the $e_i \geq 1$. It is sufficient to split n into two non-trivial factors which can then be tested for primality since we may repeat the process until we obtain the prime factorisation of n.

Asymptotically, the fastest known factoring algorithm for splitting integers is the number field sieve, it is also the fastest known algorithm in practice for factoring integers with greater than approximately 110 digits [9].

1.1 Integer factorisation

There are two principal reasons for factoring integers and we will consider both. The first class of integers is those that are in themselves interesting or special in some way such that finding the prime decomposition is desirable. The second class of integers is those that are used in cryptography.

1.1.1 Integers with known special form

Usually, for a number or a set of numbers to hold some special interest (not resulting from a real world application) we would expect to have some information regarding the properties or form of the numbers. There are many examples but there are some numbers with a special form or property that we are interested in factoring for historical reasons, pure curiosity or other non-cryptographically motivated reasons. These include, but are not limited to, Fermat and Cunningham numbers, Mersenne numbers, Cullen and Woodall numbers, repunits and cyclotomic numbers. There are many ongoing projects devoted to such computations, utilising both general and specialist factorisation algorithms, a great many of these are listed by the *World Integer Factorisation Center* [69].

Prior to the advent of public-key cryptography (which increased the importance of research into factoring general integers) the vast majority of numbers we wished to factor would be considered to have some special form. We might have practical reasons that drive the desire to factor a number or in other cases the factorisation of specific examples simply adds to the body of knowledge about numbers of the type concerned. The desire to factor these numbers provided the main impetus for research into improved and new factoring algorithms. Algorithms were tested or show cased using numbers generally regarded as being "harder" to factor and in particular the Fermat and Cunningham numbers fulfilled a key role. Eventually it was the special form of these numbers that triggered the discovery of what is now the fastest general factoring algorithm — the number field sieve (NFS).

The original paper on the number field sieve [62] described a special form — which is asymptotically faster than the later general case — for numbers of a specific known special structure. This special case encompassed both Fermat and Cunningham numbers. In these cases and later for other special forms we are able to select particularly good parameters by hand for NFS and these cases have become collectively known as the special number field sieve (SNFS).

By the time the number field sieve was invented there was a more pressing reason to attempt to improve factoring algorithms: it appeared that behind one of the foundations of secure communication lay the question "What are the current limits of what we can factor?". Numbers with *known* special form should not be used in cryptography but the factorisation of these numbers is still of interest, since

they may provide insights that lead to advances in general factoring methods.

The special form of Fermat and Cunningham numbers was partially responsible for the advent of the number field sieve. Fermat Numbers were first considered in 1640 by, and are named for, Pierre de Fermat. Fermat noted that a number of the form $2^k + 1$, $k \in \mathbb{Z}$, $k > 0$ can only be prime if k is a power of 2. The Fermat numbers are defined to be numbers $F_k = 2^{2^k} + 1$, $k \in \mathbb{Z}$, $k \geq 0$ and a Fermat prime is any number of this form which is prime. Fermat believed that all Fermat numbers were prime and indeed $F_0, \ldots, F_4$ are, however no other Fermat prime is currently known and it is now considered likely that there is only a finite quantity of Fermat primes.

In 1732 Euler [43] produced the first factorisation of a Fermat number, F_5, but gave no information regarding how this result was obtained (although later publications might suggest a possible method). F_9 was the first important factorisation result obtained using the number field sieve [62], the factorisation being used to showcase the new algorithm and to demonstrate its power when applied to numbers of this form.

During the nineteenth century many mathematicians became interested in the factorisation of Fermat numbers and numbers of the more generalised form $2^k \pm 1$ and this interest was extended to what is now referred to as the class of Cunningham numbers. These numbers are named for Lt. A. J. Cunningham who factored many of them during his lifetime. More importantly, he and Woodall collected together the first tables [31] of factorisations of numbers of the form $b^k \pm 1$, b small, k large. The interest in factoring numbers of this form is retained to this day and *The Cunningham Project* [90] is thought to be the longest running computation in the world. The original tables compiled by Cunningham and Woodall have been updated several times in book form [11] and additional updates can be found on the Cunningham project website [90]. The project has expanded to take in larger values of b as detailed in [10].

The number field sieve plays a key role in the continued production of these tables not least because it is asymptotically faster on Fermat numbers, Cunningham numbers and others with similar special form. Additionally, a more general form of the number field sieve is sometimes required to factor composite cofactors of Cunningham numbers for which no special form can be used.

While we are interested in factoring special numbers both to drive forward research, to add to the current body of knowledge and occasionally because the number is important in another branch of mathematics, the main driving force behind research into factoring algorithms is the importance of public-key cryptography to the modern electronic world.

1.1.2 General integers

The advent of public-key cryptography solved some fundamental problems faced when attempting to communicate securely. The first publication (parts of public-key cryptography appear to have been known previously in government circles) in the area came in 1976 when Diffie and Hellman [35] gave an abstract way of providing secure communication between two people who had not met or exchanged securely a secret key. This would effectively solve the key distribution problem. A practical public-key and signature cryptosystem came in 1978 due to Rivest, Shamir and Adleman [84] who published the algorithm that would become known as RSA.

The Basic Principles of the Original RSA Algorithm:

Let $n = pq$ where p and q are two primes of approximately the same size (though not too close together) and sufficiently large as to prohibit factorisation of n in a realistic time (except by luck!). Let e and d be integers such that $ed \equiv 1 \bmod \phi(n)$ where $\phi(n) = (p-1)(q-1)$ is Euler's function. Then n is referred to as the RSA modulus, e as the encryption exponent and d as the decryption exponent. The pair (n, e) is referred to as the public key and the pair (n, d) as the private key. Let m be a block of plaintext and $\gcd(m, n) = 1$. We encrypt m thus:

$$c \equiv m^e \bmod n,$$

and decrypt c:

$$m \equiv c^d \bmod n.$$

This results in the recovery of the plaintext. The details are not included as we are primarily interested in the question of how factoring the modulus results in

a complete break of the algorithm. The original paper [84] may be consulted for the details.

If we have access to p and q (by, for instance, factoring the modulus) then it is a simple matter to produce $\phi(n) = (p-1)(q-1)$. Since we also have e (as it forms part of the public key) we are then able to compute d with relative ease, as

$$d = e^{-1} \bmod \phi(n)$$

and therefore gain access to the private key. Thus factoring is sufficient to break RSA. It is also the case that given d and the public key (n, e) we may efficiently factor n. The details of this and various other attacks are contained in a summary of attacks on RSA given by Boneh [8]. It is not known whether it is necessary to factor the modulus n to efficiently compute e^{th} roots modulo n and so gain access to the plaintext. It is sometimes possible to recover plaintext without finding d (for details see Crouch and Davenport [30]).

Large integers became important cryptographically with the advent of public-key cryptography. RSA and its variants made knowledge of what size of integer we may factor, given a specific set of resources, particularly important. Hence, in order to aid us in selecting appropriately sized parameters for RSA we continue to conduct research into general factoring algorithms.

1.2 A family of algorithms

One family of factoring algorithms relies on the same intrinsic idea and the number field sieve is currently the asymptotically fastest of this group. The idea is that of factorisation by congruent squares and has its roots in a method of factoring used by Fermat. Fermat looked for x and y such that $x^2 - y^2 = n$. If such x and y could be found then this guarantees that we may produce a non-trivial factor of n by calculating $\gcd(x - y, n)$. However, finding such x and y by trial and error is not generally an easy task. As noted by Pomerance [83], Gauss and Seelhoff also used this idea to factor integers and expanded on it.

1.2.1 Factorisation by the congruent squares method

In the 1920s Kraitchik [82, 83] noted that we may relax the condition that $x^2 - y^2 = n$ and instead look for two squares which are only congruent modulo n. This no longer guarantees a non-trivial factor but makes it far easier to find values of x and y.

Suppose for n not a prime power, we construct several pairs x, y such that

$$x^2 \equiv y^2 \bmod n.$$

If in addition we have $x \not\equiv \pm y \bmod n$ we then find

$$\begin{aligned} x^2 \equiv y^2 \bmod n &\Rightarrow n \mid (x-y)(x+y), \\ x \not\equiv \pm y \bmod n &\Rightarrow n \nmid (x-y),\ n \nmid (x+y), \end{aligned}$$

thus $\gcd(n, x-y)$ and $\gcd(n, x+y)$ are non-trivial factors of n.

We may exploit this method by attempting to construct random or pseudo-random pairs $x \bmod n$, $y \bmod n$. It can be shown that if n is divisible by at least two distinct odd primes then at least half of such pairs (x, y) will produce a non-trivial factorisation of n.

This change is important as it provides a better method of producing the pairs x and y. As summarised by Pomerance [83], Kraitchik considered the polynomial $Q(a) = a^2 - n$, and for a set of integers a_i, where each a_i^2 is close to n, calculated $Q_i = Q(a_i)$. He then attempted to factor each Q_i in the hope that a subset of the Q_i could be produced that, when multiplied together, produced a square, say y^2. Let the product $a_1 \ldots a_k$ be denoted x. Then we have

$$\begin{aligned} x^2 = a_1^2 \ldots a_k^2 &\equiv (a_1^2 - n) \ldots (a_k^2 - n) \\ &\equiv Q_1 \ldots Q_k \\ &\equiv y^2 \bmod n \end{aligned}$$

It is then hoped that $x \not\equiv \pm y \bmod n$; if not then further pairs would have to found.

Later, Lehmer and Powers (1931) used a similar method that utilised continued fractions to produce different (and generally smaller) numbers Q_i that are congruent modulo n to squares. This method was abandoned due to the fact that large amounts of hand computation would often lead only to failure [11, 74].

In fact, both of these methods had two significant downsides:

1. We need to factor the "auxiliary" numbers Q_i in order to produce the required squares. This in itself is time consuming and it is difficult to decide when to abort attempts to factor a number in order to reduce the cost of this step.
2. There was no systematic approach by which to isolate a subset of the Q_i so that we may produce the required squares.

The problem of the cost of factoring the auxiliary numbers was addressed subsequently in a manner which had its roots in far earlier sieving methods, this will be discussed presently. In the interim we make a detour and consider a reduction of the first problem and the solution of the second problem. This was addressed prior to the rise of the modern sieving methods with the introduction of a non-sieving algorithm, the continued fraction method (CFRAC).

The Continued Fraction method, CFRAC

It was Morrison and Brillhart in 1970 [74] who developed Lehmer and Powers' earlier method into the first of what we might consider to be the modern factoring algorithms based on the congruent squares method. They addressed both the problem of when to abort attempts to factor the auxiliary numbers and produced a systematic approach by which the required subsets could be constructed. This was achieved by the introduction of two important components: firstly they used a "factor base", a set of prime numbers below a certain bound B and considered only integers with a similar function to the Q_i that factored entirely over this factor base; secondly they gave a method by which linear algebra could be used to find the required set.

Morrison and Brillhart demonstrated the power of the continued fraction method

by factoring the seventh Fermat number (F_7) which was at the time one of the most wanted factorisations in the Cunningham project. The continued fraction method was not a sieving method, and in fact it would see the end of the use of older sieving methods for factorisation until 1982 with the advent of the quadratic sieve, the immediate forerunner of the number field sieve.

1.2.2 A framework for algorithms of this form

Definition 1 *An integer x is said to be B-smooth if every prime factor of x is at most B.*

A general factoring algorithm from the family of congruent squares methods has three parts (with acknowledgements to [62, page 326]):

- *Select the factor base:* Select a finite set, called a factor base, of primes $p_i \leq B$. We assume that the elements of the factor base have multiplicative inverses modulo n else we immediately find a non-trivial factorisation of n.
- *Collect relations between elements of the factor base:* We collect relations between the p_i, that is we find squares x_j^2 that are B-smooth:

  $$x_j^2 \equiv \prod_{i \in I} p_i^{v_{ij}} \bmod n.$$

  We require the set of relations of this type to have slightly more elements then the factor base. We write each relation as a vector $v_j = (v_{1j}, \ldots, v_{ij}, \ldots)$ of the exponents.
- *Finding dependencies:* For each vector v_j we find $\overline{v_j}$ by reducing each of the coordinates modulo 2. We then form a matrix M by taking the $\overline{v_j}$ as columns. Since there are more relations than factor base elements the columns of the matrix are linearly dependent and we wish to find dependencies modulo 2. This is equivalent to finding a vector in the nullspace of the matrix.

  The vector will define a subset S of the relations such that $\sum_{v_j \in S} \overline{v_j} = 0$, that is, each coordinate of $v = \sum_{v_j \in S} v_j = (v_1, \ldots, v_i, \ldots)$ will be even and

hence

$$\begin{aligned}\prod_{x_j \in S} x_j^2 &\equiv \prod_{i \in I} p_i^{v_i} \\ &\equiv y^2 \bmod n\end{aligned}$$

Since we have two squares congruent modulo n the two square method above will lead to a non-trivial factorisation of n with probability just over one half. Ten such dependencies will produce a non-trivial factorisation of n with high probability.

CFRAC was a significant step forward as it introduced both the factor base and linear algebra in the above scheme and it is this that allows us to produce algorithms utilising the two squares method since it provides a general approach by which the congruent squares may be produced.

The quadratic sieve:

The quadratic sieve took parts of CFRAC, Kraitchik's method and older sieving methods in order to produce a sophisticated algorithm. Pomerance first became interested in this amalgamation of ideas when he noticed that the theoretical runtime of such an algorithm was more favourable than CFRAC, and described the algorithm in [82]. We give a basic description:

We consider the polynomial $Q(a) = (\lfloor\sqrt{n}\rfloor + a)^2 - n$. Clearly $Q(a) \equiv x^2 \bmod n$ for $x = (\lfloor\sqrt{n}\rfloor + a)$.

The factor base will consist of prime numbers below $B \in \mathbb{N}$, the factor base bound. We need not include all the primes p below B, indeed if p is odd p can only divide $Q(a)$ if the Legendre symbol $(\frac{n}{p}) = 1$.

We wish to find values of a for which $Q(a)$ is B-smooth — i.e. $Q(a)$ factors completely over the factor base. However, trial factoring these over the factor base would be costly and many would not factor. Instead, since $Q(a)$ is a polynomial with integer coefficients we use a sieve. This is the major difference between CFRAC and the quadratic sieve.

Sieving dates back to the sieve of Eratosthenos (a way of finding the prime numbers in an interval). The quadratic sieve is based on the following idea: if we have some $m \in \mathbb{Z}$ and we know that $m|Q(a)$ then we can immediately deduce that $m|Q(a+km)$, $\forall k \in \mathbb{Z}$. Hence once we have an a such that $m|Q(a)$ we can cheaply identify others.

For each prime p in the factor base (excepting $p = 2$ which is a special case) we find two roots of $Q(a) \equiv 0 \bmod p$ and call these r_1, r_2. To isolate the B-smooth $Q(a)$ for some a interval we find the first sieve locations a for which $a \equiv r_i \bmod p$ and then we divide each $Q(a+kp)$ by p. After we have processed all p in the factor base those values of $Q(a)$, across the interval, which have been reduced to 1 are B-smooth (this will not find all the B-smooth numbers, for instance it will fail for any number divisible by 2^2 so the actual process is somewhat more complicated).

Replacing $Q(a)$ in $Q(a) \equiv x^2 \bmod n$ by the prime factorisations, we see that each a gives rise to a multiplicative relation (modulo n) between elements of the factor base. As described above these may then be written as relations viewed as vectors of exponents.

We continue sieving until there are slightly more relations than elements in the factor base. We may then use linear algebra as described in the general framework above to produce dependencies and hence squares x^2 and y^2 congruent modulo n.

Choosing a good smoothness bound is of great importance in the above algorithm. If the bound B is too small we may never find enough $Q(a)$ that factor over the factor base. If the bound B is too large then the factor base is also large and we will have to find a great many relations and hence this may produce a matrix that is too large to handle in final step.

The quadratic sieve was quickly found to be practical and then easily outperformed CFRAC not least due to methods that allowed multiple polynomials to be used [82, 86]. Indeed the multiple polynomial quadratic sieve (MPQS) remains a practical algorithm for numbers up to approximately 110 digits [9] (where the crossover point to NFS occurs) that are not more suited to other factoring algorithms such as the elliptic curve method [47].

The quadratic sieve is an immediate precursor to the number field sieve which was first introduced as a special case — a generalisation of the quadratic sieve that appeared only to be relevant for Fermat and Cunningham numbers. The process of producing a viable algorithm for general integers was not a simple one and will be detailed below. However, as we will see, the resulting algorithm proved to be significantly more powerful than MPQS.

For further information on the history of factoring algorithms based on the congruence of two squares see [11, 74, 82, 83]. For a survey on modern factoring algorithms see [9].

Complexity

If $\mathcal{F}(B)$ is our factor base then an algorithm of the general form above requires us to find approximately $|\mathcal{F}(B)|$ B-smooth numbers. *If* the numbers that we test for smoothness were randomly selected positive integers up to x then each one is B-smooth with probability $\Psi(x,B)/x$, where $\Psi(x,B)$ is the quantity of B-smooth numbers in the interval $[1,x]$. In this case we would expect to need to test $x|\mathcal{F}(B)|/\Psi(x,B)$ numbers in order to find the required quantity of relations.

If we were to use trial division, as we do in CFRAC, in order to verify that a number was B-smooth it would take about $|\mathcal{F}(B)|$ steps hence we would expect to take $x|\mathcal{F}(B)|^2/\Psi(x,B)$ steps in all. In CFRAC the numbers we test for smoothness are positive integers of size up to $O(n^{1/2+\epsilon})$.

In the quadratic sieve the numbers that we test for smoothness are also of size up to $O(n^{1/2+\epsilon})$, however, we do not trial divide but use a sieving process in order to verify that numbers are B-smooth. As described above, a sieving process allows us to cheaply identify a set of integers divisible by some integer m, for instance, once we have identified one integer divisible by m. This in turn permits us to check sets of numbers for B-smoothness without trial dividing every number by all primes below B. This takes about $\log\log B$ steps on average for each number in the set and it is this that is responsible for the quadratic sieve outperforming CFRAC.

For some factoring algorithms an argument along these lines may be used as a

route to a rigorous complexity analysis, however, for others we must make various heuristic assumptions. In particular we often need to make an assumption that the numbers we test for smoothness are as likely to be smooth as random integers of the same size.

1.3 The number field sieve

The number field sieve (NFS) is an algorithm that follows the general pattern given above. It is currently the fastest factoring algorithm for numbers greater than approximately 110 digits and it is also asymptotically fastest. The idea behind the special number field sieve (SNFS) was first introduced by Pollard [80]. The original special number field sieve, as described fully in [63], is a factoring algorithm for numbers with a specific form:

$$n = r^e - s$$

or small integer multiples thereof, where r, $|s| \in \mathbb{N}$ are small and e is large. Examples of numbers of this form include Fermat numbers and Cunningham numbers. The first significant example of factoring using SNFS is the factorisation of the ninth Fermat number in [62]. In this case the algorithm is in its simplest case due to the properties of that particular number.

SNFS cannot be used to factor numbers that are generally used in cryptography. The algorithm was generalised in [12], although certain practical issues were not resolved and no factorisations were completed at this point. It took the work of many others to address a myriad of practical concerns and there are still areas where significant improvements need to be made. This algorithm is now known as the general number field sieve (GNFS). In this thesis where no distinction between SNFS and GNFS is necessary, they will be referred to jointly as the number field sieve (NFS).

A brief description of the algorithm

For necessary background we refer readers to [21, 33, 59, 88].

Suppose we select two irreducible polynomials over the integers $f_1(X)$ and $f_2(X)$ of degree d_1 and d_2 respectively, for which there exists a common root m mod n. Let $\alpha_1, \alpha_2 \in \mathbb{C}$ be such that $f_i(\alpha_i) = 0, i = 1, 2$ and define two number fields $K_i = \mathbb{Q}(\alpha_i), i = 1, 2$. We produce a square in each number field and define two homomorphisms $\mathbb{Z}[\alpha_i] \rightarrow \mathbb{Z}/n\mathbb{Z}$ by sending each α_i to m, by which means we may then produce $x, y \in \mathbb{Z}$ such that $x^2 \equiv y^2 \bmod n$ as required.

The squares are produced by considering the values of the homogeneous polynomials $F_i(X, Y) = Y^{d_i} f_i(X/Y)$. We find pairs (a, b) such that a, b are coprime and the auxiliary numbers $F_1(a, b)$, $F_2(a, b)$ are B_1 and B_2 smooth respectively, that is, they factor over the primes below some user defined factor base bound B_i. These pairs are the relations in the number field sieve and they are found using a sieving process.

Once a sufficient quantity of relations have been found they are filtered, if required, to reduce the quantity of raw data, and a matrix is formed. Finding the required squares is then a case of establishing linear dependencies among the columns of the matrix.

Having produced the squares in the number fields we then require the images of their square roots in $\mathbb{Z}/n\mathbb{Z}$. This is not without difficulty. Each dependency produces one pair of such integers and for each pair the probability of producing a non-trivial split of n is at least one half.

The number field sieve can be split into four distinct steps:

1. Selection of polynomials.
2. Sieving for relations.
3. Filtering raw data and linear algebra.
4. Finding the square roots in the number fields.

Great advances have been made in all of these steps although in the general case the first and third steps are less well developed.

The number field sieve outperforms the quadratic sieve as the numbers that we

wish to test for smoothness are significantly smaller than $O(n^{1/2+\epsilon})$. In fact x will be bounded by an expression of the form $\exp((\log n)^{2/3}(\log\log n)^{1/3})$. This ensures that the number field sieve has asymptotic runtime of the form $\exp((c+o(1))(\log n)^{1/3}(\log\log n)^{2/3})$, c constant, while that of the quadratic sieve is $\exp((1+o(1))(\log n)^{1/2}(\log\log n)^{1/2})$. It is the combination, in the number field sieve, of significantly smaller auxiliary numbers and the use of sieving to determine whether they are smooth which results in an far more advanced and powerful algorithm.

1.4 Contribution of this thesis

We consider a method of estimating the quantity of relations produced by the number field sieve worked on by Boender, Murphy and Cavallar. We examine methods by which we may produce a more dependable estimate for the quantity of relations produced by the full algorithm and provide empirical evidence to support this. Such estimation methods are not only of use when attempting to select parameters for the number field sieve but also when comparing variants of the algorithm. We use this estimation method to aid our analysis of a variant of the (special) number field sieve in a later chapter.

We note a collection of characteristics of the special cases of the number field sieve and in particular the presence of subfield structure in some of the number fields utilised in these cases.

We consider the (human selected) special cases of the number field sieve in the light of certain advances made in the general case. In particular we work with a collection of polynomial selection methods produced by Murphy which have been shown to produce substantially improved results in general, and we find that using these we are able to reproduce certain aspects of the special cases without reference to "specialness". We note a loss of distinction between the special and general cases of the number field sieve and consider the implications that this has, posing some open questions. In particular we examine the possibility of a new repudiation attack on RSA. This work suggests that we need to consider variants of the special number field sieve with more care, as it is possible for automated methods to identify some special cases (without knowledge of any

special structure present) that may not have been recognised as such by a human and may have been used for cryptographic purposes. The density of integers which may be automatically found to be special cases for the number field sieve cannot be easily quantified — although we might suppose that they are rare in some sense or perhaps that there is a continuum of NFS "hardness"; this is an open question.

We consider a method of utilising the subfield structure found in various special cases. With the aid of theory, sieving tests and estimation of the quantity of data produced we show that while implementation of this method is possible it is not of any immediate practical benefit.

1.5 Outline of this thesis

In chapter 2 we give an exposition of the number field sieve. In chapter 3 we cover necessary theoretical background. We will refer to this material throughout the subsequent chapters.

Chapter 4 focuses on improvements to the estimation of the yield of the sieving step, which we use in chapter 6. In chapter 5 we provide an in depth survey of the characteristics of the known special cases of the number field sieve, comparing and contrasting these with the general case. In chapter 6 we consider a natural variant of the number field sieve that uses subfield structure and provide strong evidence that this variant is unlikely to prove useful in its current form. In chapter 7 we consider the polynomial selection methods that are currently used in the general case and seek to place the special cases of chapter 5 in context.

Chapter 8 contains a summary and suggestions for further work.

Chapter 2

Background: The number field sieve

2.1 The general number field sieve

The general number field sieve was described in great detail by Buhler, H. Lenstra and Pomerance [12]. A more modern description due to Huizing et al. can be found in [18, 39]. We outline the algorithm below.

We choose two irreducible polynomials over the integers, $f_1(X)$ and $f_2(X)$ of degree d_1 and d_2 respectively, for which there exists an integer m such that

$$f_1(m) \equiv f_2(m) \equiv 0 \bmod n.$$

To simplify the explanation we will assume that the f_i are monic, a restriction which will later be lifted. Let $\alpha_i \in \mathbb{C}, i = 1, 2$ be such that $f_i(\alpha_i) = 0$ and define two number fields $K_i = \mathbb{Q}(\alpha_i)$. We may now define two ring homomorphisms:

$$\begin{aligned} \varphi_i : \mathbb{Z}[\alpha_i] &\rightarrow \mathbb{Z}/n\mathbb{Z} \\ \varphi_i(\alpha_i) &\equiv m \bmod n \\ \varphi_i\left(\sum_{j=0}^{d_i-1} a_j \alpha_i^j\right) &\equiv \sum_{j=0}^{d_i-1} a_j m^j \bmod n. \end{aligned}$$

We wish to construct a set S of integer pairs (a, b) with, a coprime to b for which

$$\prod_{(a,b)\in S} (a - b\alpha_1) = \beta_1^2 \in \mathbb{Z}[\alpha_1]$$

and

$$\prod_{(a,b)\in S} (a - b\alpha_2) = \beta_2^2 \in \mathbb{Z}[\alpha_2]$$

Since ring homomorphisms preserve multiplication, that is

$$\varphi(\theta_1\theta_2) = \varphi(\theta_1)\varphi(\theta_2), \ \ \forall\, \theta_1, \theta_2 \in \mathbb{Z}[\alpha]$$

we have

$$\begin{aligned}
\varphi_1(\beta_1)^2 &\equiv \varphi_1(\beta_1^2) \\
&\equiv \prod_{(a,b)\in S} \varphi_1(a - b\alpha_1) \\
&\equiv \prod_{(a,b)\in S} (a - bm) \\
&\equiv \prod_{(a,b)\in S} \varphi_2(a - b\alpha_2) \\
&\equiv \varphi_2(\beta_2^2) \equiv \varphi_2(\beta_2)^2 \bmod n.
\end{aligned}$$

Thus we produce two squares congruent modulo n as required. We may then calculate $\gcd(n, \varphi_1(\beta_1) - \varphi_2(\beta_2))$ which, assuming that n has at least two distinct, odd prime divisors, will produce a non-trivial factor of n in at least half the cases.

In order to find such a set S we work in a similar manner to that described in the preceding chapter. However, in this case we are interested in finding pairs (a, b) such that the $a - b\alpha_i \in \mathbb{Z}[\alpha_i]$ are smooth in some sense. In order to do this we need both an idea of smoothness in $\mathbb{Z}[\alpha_i]$ and a method of finding those elements that are smooth. We will define smoothness in terms of the norm of an algebraic integer, for background and more general definitions we refer the reader to [21, 59, 88]:

Definition 2 *An algebraic integer $\beta \in Z[\alpha]$ is said to be B-smooth if the absolute value of its norm, $|\mathbf{N}(\beta)|$ is B-smooth in the usual sense.*

The norm of an algebraic number $\beta \in \mathbb{Q}(\alpha)$ is defined to be

$$\mathbf{N}(\beta) = \prod_{i=1}^{d} \sigma_i(\beta)$$

for $\mathbb{Q}(\alpha)$ and f both of degree d, where σ_i are the embeddings of $\mathbb{Q}(\alpha)$ into the complex numbers. However, we do not need to consider a general definition of the norm as the specific case that we are working in provides us with a more accessible definition. We associate with each f_i a homogeneous polynomial

$$F_i(X, Y) = Y^{d_i} f_i(X/Y).$$

Recalling that the f_i are monic we have, from [49] for example, that

$$\mathbf{N}(a - b\alpha_i) = F_i(a, b).$$

We find coprime pairs (a, b) such that both $F_1(a, b)$ and $F_2(a, b)$ are smooth. Utilising these we wish to produce a product of the form $\prod_{(a,b)\in S}(a - b\alpha_i)$ which we know to be a square in $\mathbb{Z}[\alpha_i]$. It is necessary for the norm $\prod_{(a,b)\in S} F_i(a, b) = \mathbf{N}(\prod_{(a,b)\in S}(a - b\alpha_i))$ to be a square in $\mathbb{Z}$ to ensure that $\prod_{(a,b)\in S}(a - b\alpha_i)$ is a square in $\mathbb{Z}[\alpha_i]$ but it is not sufficient: we need to pay more attention to the type of prime that can divide the norm.

We work identically on both sides and hence, dropping subscripts for the moment, suppose we have f of degree d, $f(\alpha) = 0$ and $K = \mathbb{Q}(\alpha)$. We will assume that $\mathbb{Z}[\alpha]$ is equal to the ring of integers $\mathcal{O}$, this is a strong assumption that will not usually hold, but later we will be able to relax the assumption. Under this assumption we are working in a Dedekind domain and have unique factorisation into ideals, we recall the following necessary facts:

- The norm of the ideal generated by β, written $\mathfrak{N}\langle\beta\rangle$ is equal to the norm $\mathbf{N}(\beta)$.
- For every non-trivial prime ideal $\mathfrak{p}$ of $\mathcal{O}$, $\mathfrak{N}\mathfrak{p} = p^k$, some $k \in \mathbb{N}$. k is said to be the degree of the ideal.
- For ideals generated by rational primes we may write $\langle p\rangle = \prod \mathfrak{p}_i^{e_i}$ where the exponents are positive integers.

We are interested in algebraic integers of a special form $a - b\alpha$. Let $e_p(a - b\alpha) = \mathrm{ord}_p \mathbf{N}(a - b\alpha)$ be the number of times that p divides the norm $\mathbf{N}(a - b\alpha)$. We may write

$$\begin{aligned} F(a,b) &= \mathbf{N}(a - b\alpha) = \prod_p p^{e_p(a-b\alpha)} \\ \mathfrak{N}\langle a - b\alpha \rangle &= \mathfrak{N}\left(\prod_{\mathfrak{p}} \mathfrak{p}^{v_{\mathfrak{p}}(a-b\alpha)} \right) \end{aligned}$$

where $v_{\mathfrak{p}}(a-b\alpha)$ is the $\mathfrak{p}$-adic valuation of $a-b\alpha$. Finally we have that $\mathbf{N}(a-b\alpha) = \mathfrak{N}\langle a - b\alpha \rangle$. We are concerned with the situation where $\prod_p p^{e_p(a-b\alpha)}$ is known to be a square. Since we are working in a Dedekind domain, $\prod_{\mathfrak{p}} \mathfrak{p}^{v_{\mathfrak{p}}(a-b\alpha)}$ is a square if and only if all the $v_{\mathfrak{p}}(a - b\alpha)$ are even. Hence it is necessary for the $e_p(a - b\alpha)$ to be even for $\prod_{\mathfrak{p}} \mathfrak{p}^{v_{\mathfrak{p}}(a-b\alpha)}$ to be a square but not sufficient — there are two ways in which the e_p could all be even while the $v_{\mathfrak{p}}$ are not all even:

- $\mathfrak{N}\mathfrak{p} = p^k$ for some $k > 1$,
- $p \in \mathbb{Z}$, p prime is contained in two or more distinct prime ideals.

The first issue will not arise due to the special form of our algebraic integers. However the second issue requires us to consider more carefully what type of p divide the norm.

More specifically, the first issue will not occur due to the following lemma which is proved in [62]:

Lemma 1 *Let $a, b \in \mathbb{Z}$ with $\gcd(a, b) = 1$. Then every prime ideal $\mathfrak{p}$ that occurs in $a - b\alpha$ is a first degree prime ideal.*

To overcome the second issue, we must heed what kind of p divide the $F(a, b)$. For each prime p below the smoothness bound B we define the set

$$\mathcal{R}(p) = \{r \in \mathbb{Z}/p\mathbb{Z} \mid F(r, 1) \equiv 0 \bmod p\}.$$

As we shall see we then have for coprime a and b that $F(a, b)$ is divisible by p if

and only if $a \equiv br \bmod p$ for $r \in \mathcal{R}(p)$ hence we consider our factor base to be

$$\mathcal{F}(B) = \{(p, r) \mid p \text{ prime}, p < B, r \in \mathcal{R}(p)\}.$$

In fact we note a lemma [12, 63]:

Lemma 2 *If $p \in \mathbb{Z}$, p prime and $\mathcal{R}(p)$ as above then there is a one to one correspondence between pairs (p, r) with $r \in \mathcal{R}(p)$ and the first degree prime ideals $\mathfrak{p}$ of $\mathcal{O}$ $(= \mathbb{Z}[\alpha])$.*

There may be more than one r for each p.

Thus if $\mathfrak{p}$ corresponds to (p, r) we have $\mathfrak{N}\mathfrak{p} = p$ and the map $\mathbb{Z}[\alpha] \to \mathbb{Z}[\alpha]/\mathfrak{p} \cong \mathbb{Z}/p\mathbb{Z}$ maps α to $r \bmod p$. $\mathfrak{p}$ is generated by p and $r - \alpha$. We may use this map to test if an element of $\mathbb{Z}[\alpha]$ is contained in $\mathfrak{p}$:

$$\sum_{j=0}^{d-1} a_j \alpha^j \in \mathfrak{p} \iff \sum_{j=0}^{d-1} a_j r^j \equiv 0 \bmod p.$$

For the elements that we wish to work with we now have

$$a - b\alpha \in \mathfrak{p} \iff a - br \equiv 0 \bmod p.$$

This implies that we will require that each $e_{(p,r)}(a - b\alpha)$ be even instead of just each $e_p(a - b\alpha)$. We now have an exact correspondence:

$$F(a, b) = \mathrm{N}(a - b\alpha) = \mathfrak{N}\left(\prod_{\mathfrak{p}} \mathfrak{p}^{v_{\mathfrak{p}}(a-b\alpha)}\right) = \prod_{p} p^{e_{(p,r)}(a-b\alpha)}$$

and $v_{\mathfrak{p}}(a - b\alpha) = e_{(p,r)}(a - b\alpha)$.

We can now construct a set S:

- The factor bases $\mathcal{F}(B_1), \mathcal{F}(B_2)$ consist of all the first degree prime ideals of $\mathbb{Z}[\alpha_1]$ with norm at most B_1 and of $\mathbb{Z}[\alpha_2]$ with norm at most B_2 respectively. These are in one to one correspondence with the pairs $(p, r)_i$, $i = 1, 2$. As $B_i \to \infty$ the size of the factor base is approximately $\pi(B_i)$, the number of primes below B_i, by the Chebotarev density theorem [58, section VIII,4].

- We collect pairs (a, b) such that $a - b\alpha_1$ and $a - b\alpha_2$ are smooth by sieving over the values $F_1(a, b)$ and $F_2(a, b)$ respectively. We check that a, b are coprime, $a - br \equiv 0 \bmod p$, for $(p, r)_i$, $i = 1, 2$. Each smooth pair forms a relation. We collect slightly more relations than there are elements in the factor base.

- In addition to the sieved data we have the so called free relations (which are valid prior to applying the ring homomorphisms and can be found essentially for free when computing the factor base). One free relation is found for each $p \leq \min(B_1, B_2)$ for which both f_1 and f_2 factor completely into distinct linear factors modulo p.

  If $\mathrm{Gal}(f_1 f_2)$ is the Galois group of $f_1 f_2$ then approximately $1/|\mathrm{Gal}(f_1 f_2)|$ of the set of primes will produce a free relation as $B_i \to \infty$ (excepting those that are not squarefree). Thus we would like to use f_i that minimise $\mathrm{Gal}(f_1 f_2)$ if possible [39].

- We use linear algebra over GF(2) to find a set S of (a, b) such that

$$\sum_{(a,b)\in S} e_{(p,r)_i}(a - b\alpha_i) \equiv 0 \bmod 2,$$

  for all primes in the factor bases.

However this is not strong enough to ensure that the set S satisfies

$$\prod_{(a,b)\in S} (a - b\alpha) \in \mathbb{Z}[\alpha] \text{ is a square.}$$

Before closing this gap we will show that we can retain the correspondence between the factorisation of $F(a, b)$ and factorisation of ideals of orders other than $\mathcal{O}$.

Relaxing our assumptions:

Working identically for $i = 1, 2$ we drop the subscripts. We relax the assumption that the order in which we work is the ring of integers of $\mathbb{Q}(\alpha)$. We need to connect the ideal structure of the order in which we must work, A, with the known structure of the maximal order $\mathcal{O}$ and hence retain the correspondence

between the norm factorisation of $F(a,b)$ and the ideal factorisation in A. If f is monic we work in $A = \mathbb{Z}[\alpha]$.

We have a result from [12] which introduces homomorphisms $l_{\mathfrak{p}}$; if $A = \mathcal{O}$ then $l_{\mathfrak{p}}(\beta)$ is the exponent of the power of $\mathfrak{p}$ dividing the ideal $\beta\mathcal{O}$. It is possible to think of the following as a generalisation of $\mathfrak{p}$-adic valuations to the order $A = \mathbb{Z}[\alpha] \neq \mathcal{O}$.

Proposition 1 *There exists for each prime $\mathfrak{p}$ of $\mathbb{Z}[\alpha]$ a group homomorphism*

$$l_{\mathfrak{p}} : K^* \to \mathbb{Z}$$

such that the following hold:

1. $l_{\mathfrak{p}}(\beta) \geq 0$, $\forall\, \beta \in \mathbb{Z}[\alpha]$, $\beta \neq 0$.
2. *If $\beta \in \mathbb{Z}[\alpha]$, $\beta \neq 0$ then $l_{\mathfrak{p}}(\beta) > 0$ if and only if $\beta \in \mathfrak{p}$.*
3. $\forall\, \beta \in K^*$, $l_{\mathfrak{p}}(\beta) = 0$ *for all but finitely many $\mathfrak{p}$ and*

$$\prod_{\mathfrak{p}} (\mathfrak{N}\mathfrak{p})^{l_{\mathfrak{p}}(\beta)} = |\mathbf{N}(\beta)|$$

where the $\mathfrak{p}$ range over the set of all primes of $\mathbb{Z}[\alpha]$.

When $\beta = a - b\alpha$ we have a corollary [12]:

Corollary 1 *Let a,b be coprime integers and let $\mathfrak{p}$ be a prime of $\mathbb{Z}[\alpha]$. If $\mathfrak{p}$ is not a first degree prime then $l_{\mathfrak{p}}(a - b\alpha) = 0$. If $\mathfrak{p}$ is a first degree prime corresponding to the pair (p,r) then $l_{\mathfrak{p}}(a - b\alpha) = e_{(p,r)}(a - b\alpha)$.*

We now have an exact correspondence between integer factorisation of the norm $F_i(a,b) = \mathbf{N}(a - b\alpha)$ and ideal factorisation of $\langle a - b\alpha \rangle$ and hence can find a set S of coprime pairs (a,b) such that $\sum_{(a,b)\in S} e_{(p,r)}(a - b\alpha) \equiv 0 \bmod 2$ as above. However, there remain four obstructions [12] to ensuring that $\prod_{(a,b)\in S}(a - b\alpha)$ is a square in $\mathbb{Z}[\alpha]$:

1. The ideal $\prod_{(a,b)\in S}(a-b\alpha)\mathcal{O}$ of $\mathcal{O}$ may not be a square of an ideal since we work with primes of $\mathbb{Z}[\alpha]$ rather than of $\mathcal{O}$.

2. Even if $\prod_{(a,b)\in S}(a-b\alpha)\mathcal{O} = \mathfrak{a}^2$ for some ideal $\mathfrak{a} \in \mathcal{O}$ the ideal $\mathfrak{a}$ need not be principal.

3. Even if $\prod_{(a,b)\in S}(a-b\alpha)\mathcal{O} = \gamma^2\mathcal{O}$ for some $\gamma \in \mathcal{O}$ it is not necessarily the case that $\prod_{(a,b)\in S}(a-b\alpha) = \gamma^2$.

4. Even if $\prod_{(a,b)\in S}(a-b\alpha) = \gamma^2$ for some $\gamma \in \mathcal{O}$ we need not have $\gamma \in \mathbb{Z}[\alpha]$.

In summary [12] notes that: if $\mathbb{Z}[\alpha] = \mathcal{O}$, as assumed above, then obstructions 1 and 4 cannot occur. If $\mathcal{O}$ also has class number 1 and is therefore a principal ideal domain then obstruction 2 cannot occur. Finally, if $\mathcal{O}$ is a principal ideal domain and we have an explicit basis for the unit group of $\mathcal{O}$ then obstruction 3 can be handled by including a system of generating units in the factor base.

In general we are not able to make these assumptions. We are often able to make some or even all of these assumptions in SNFS. In particular, many early SNFS factorisations not only had $\mathcal{O} = \mathbb{Z}[\alpha]$ or could recover to this situation in some manner but it was possible to work with the generators of the ideals and those of the basis for the unit group. However, the method by which the above obstructions are countered is so successful that it is unusual to distinguish between the special and general cases in this part of the algorithm.

The fourth obstruction is easily countered as noted in [12]. If $\prod_{(a,b)\in S}(a-b\alpha) = \gamma^2, \gamma \in K$ then $\gamma \in \mathcal{O}$; and we have $\gamma f'(\alpha) \in \mathbb{Z}[\alpha]$ and thus $f'(\alpha)^2 \prod_{(a,b)\in S}(a-b\alpha)$ is the square of an element of $\mathbb{Z}[\alpha]$.

Quadratic characters:

To overcome the other three obstructions we use quadratic characters. This was first suggested by Adleman [1]. Other methods of circumventing these problems were attempted but these solutions require the linear algebra step to work over $\mathbb{Z}$ instead of over GF(2); this would substantially increase the cost of the linear algebra step. Using quadratic characters allows us to continue working over GF(2). From [12] we have:

Proposition 2 *Let S be a finite set of coprime integer pairs (a, b) such that*

$$\prod_{(a,b)\in S} (a - b\alpha) \text{ is the square of an element of } K = \mathbb{Q}(\alpha).$$

Let q be an odd prime number and $s \in \mathcal{R}(q)$ (as previously defined) such that

$$\begin{aligned} a - bs &\not\equiv 0 \bmod q, \ \forall (a, b) \in S \\ f'(s) &\not\equiv 0 \bmod q. \end{aligned}$$

Then we have

$$\prod_{(a,b)\in S} \left(\frac{a - bs}{q}\right) = 1$$

where $\left(\frac{x}{y}\right)$ denotes the Legendre symbol.

We are really interested in the converse of the above proposition and the converse does hold, as pointed out in [12]:

If $\beta \in \mathbb{Z}[\alpha]\backslash\{0\}$ satisfies $\chi_q(\beta) = \left(\frac{\beta}{q}\right) = 1$ for all first degree primes $\mathfrak{q}$, $q = \mathfrak{N}\mathfrak{q}$, with $2\beta \notin \mathfrak{q}$ (or even for all such $\mathfrak{q}$ with finitely many exceptions) then β is a square in K.

We use the above technology as follows: for each polynomial f_i we take several large prime ideals which are not in the factor base; that is, q odd, q prime, $q > B_i$, $s \in \mathcal{R}_i(q)$, with $(q, s)_i$ not in the factor base. We append to our relation vector 0 in the character column corresponding to $(q, s)_i$ if $\chi_q(a - b\alpha_i) = \left(\frac{a-bs}{q}\right) = 1$ and 1 if $\chi_q(a - b\alpha_i) = \left(\frac{a-bs}{q}\right) = -1$. Now we complete the linear algebra step as usual to produce a linear dependency amongst the relations. We then have

$$\chi_q\left(\prod_{(a,b)\in S} (a - b\alpha_i)\right) = \prod_{(a,b)\in S} \left(\frac{a - bs}{q}\right) = 1$$

for all of the test primes q. If there are sufficiently many q per polynomial (in [49] 32 per polynomial was deemed sufficient) then it is now almost certain that

$$\prod_{(a,b)\in S}(a - b\alpha_1) \quad \in \mathbb{Z}[\alpha_1] \text{ is a square}$$

and

$$\prod_{(a,b)\in S}(a - b\alpha_2) \quad \in \mathbb{Z}[\alpha_2] \text{ is a square.}$$

as required.

Having constructed the squares in $\mathbb{Z}[\alpha_1]$ and $\mathbb{Z}[\alpha_2]$ we must now take square roots of algebraic integers (with large coefficients) in the number field. We will consider this step and the others in more detail after disposing of one final assumption.

Throughout this section we have assumed that the f_i and the F_i are monic. This is not necessary though it simplifies explanations. Although the first methods for finding NFS polynomials produced monic polynomials, this is fairly restrictive. For instance, allowing non-monic f_i can lead to the other coefficients being smaller. It will also mean that we have the choice between using polynomials with "skewed" coefficient size and those with all the coefficients of similar size. Allowing this greater freedom only requires some minor changes.

Dropping subscripts, let $f(X) = \sum_i^d a_i X^i$ and define $F(X,Y) = Y^d f(X/Y)$, $f(\alpha) = 0$ and $K = \mathbb{Q}(\alpha)$. Choosing f monic ensures that $\alpha \in \mathcal{O}$, if we allow $a_d \neq \pm 1$ then this is no longer the case. The reason we require $\alpha \in \mathcal{O}$ is so that $\mathbb{Z}[\alpha]$ is an order. However, if we have $a_d \neq \pm 1$ we are able to show that $A = \mathbb{Z}[\alpha] \cap \mathbb{Z}[\alpha^{-1}]$ is an order [12] and we may work with A as follows:

Let ω be a zero of $F(X, a_d)$. If $\alpha = \omega/a_d$ then

$$F(\omega, a_d) = 0 \implies F(\alpha, 1) = f(\alpha) = 0$$

since F is homogeneous. Now $\mathbb{Z}[\omega]$ is an order, $\omega \in \mathcal{O}$ and $a_d(a - b\alpha) = a_d a - b\omega$. Also

$$F(a,b) = \mathbf{N}(a_d a - b\omega) = a_d \mathbf{N}(a - b\alpha)$$

c.f. $F(a,b) = \mathbf{N}(a - b\alpha)$ in the monic case.

In the monic case the first degree primes of $\mathbb{Z}[\alpha]$ are in correspondence with pairs $(p, r), r \in \mathcal{R}(p)$. Now we identify r with r_1/r_2 whenever $r_2 \neq 0$ and we define

$$\mathcal{R}'(p) = \{(r_1, r_2) \in (\mathbb{Z}/p\mathbb{Z})^2 \mid F(r_1, r_2) \equiv 0 \bmod p\} \cup \{\infty\}.$$

If $r_2 = 0$ identify $r \in \mathcal{R}(p)$ with $\infty \in \mathcal{R}'(p)$. Now let $e_{(p,r)}(a - b\alpha)$ as before, denote the exponent in $F(a, b) = \mathbf{N}(a - b\alpha)$ corresponding to the ideal (p, r), then we retain the homomorphisms $l_\mathfrak{p}$ across ideals of A, and

$$e_{(p,r)}(a - b\alpha) = \begin{cases} l_\mathfrak{p}(a - b\alpha) & \text{if } r \neq \infty \\ l_\mathfrak{p}(a - b\alpha) + \operatorname{ord}_p a_d & \text{if } r = \infty \end{cases}$$

So we may sieve again.

Now we have described the basic algorithm we need to look at how to achieve the different parts in more detail. Each of these problems have been solved to varying degrees since the advent of NFS and we will briefly describe the current situation in each.

2.1.1 Polynomial selection

There are two general methods for selecting the polynomials f_1, f_2 and various *ad hoc* methods of producing polynomials for the special number field sieve. In the general case there are more recent methods for compiling a set of candidate polynomials for a factorisation and then selecting "good" polynomials from that set.

For now we will describe only the original general polynomial construction method. The current situation of this problem will be considered in more depth in the succeeding chapters once appropriate theory has been introduced. This is one of the least developed parts of the number field sieve.

The original method for finding a general number field sieve polynomial is provided in [12]. In GNFS we know only the value n and have no knowledge of any special structure that may help us. This method is known as the "base-m" method.

1. Select a small positive integer $d_1 > 1$, $n > 2^{d_1^2}$ and set $d_2 = 1$. Optimally we have [12]

$$d_1 = \left(\frac{(3+o(1))\log n}{\log\log n}\right)^{1/3}$$

as $n \to \infty$.

2. Set $m = \lfloor n^{1/d_1} \rfloor$ and write n to the base m:

$$n = a_{d_1} m^{d_1} + a_{d_1-1} m^{d_1-1} + \cdots + a_0$$

$0 \leq a_j < m$.

3. Then $f_1(X) := a_{d_1} X^{d_1} + a_{d_1-1} X^{d_1-1} + \cdots + a_0$ and $f_2(X) := X - m$. In both cases $f(m) = n$.

We then define our number fields in the usual manner. In [12] it is noted that this method will always result in a monic polynomial f_1 with coefficient $a_{d_1-1} \leq d_1$. An argument in section 12 of the same paper argues that in a weak asymptotic sense the base-m algorithm cannot be improved on, though it is likely that for practical purposes there is still room for improvement. Later it became more usual to substitute $m \approx n^{1/(d_1+1)}$ in step two. This will produce a non-monic polynomial which, as we have just seen, is not a problem. The advantage of this change is that in general the polynomial coefficients are smaller. We may also adjust the standard base-m representation so that the coefficients have smaller absolute value but may be negative, that is, $-m/2 \leq a_j \leq m/2$ which can again result in a non-monic polynomial.

Murphy [76, section 3] provides a table of relevant d_1 for size of n. For numbers n of size between 80 and 300 digits we will use $d_1 = 4, 5,$ or 6 so we are presently only interested in degree pairs $(4,1)$, $(5,1)$ or $(6,1)$.

Various authors have suggested attempting to find polynomials with degree pair (d_1, d_2) with $d_1 \neq 1$ and $d_2 \neq 1$. However there is no apparent way of selecting such polynomials except in the case $d_1 = d_2 = 2$. A method for finding two quadratic polynomials was provided by Montgomery and is described in [49]. Further work on this method is found in [77].

From [76] we have that polynomials with degree pair $(2,2)$ may only be expected to compete with base-m polynomials with degree pair $(3,1)$. For n of size beyond

approximately 110 decimal digits variants of the base-m method are still the method of choice. The factorisations of both RSA-140 [17] and RSA-155 [18] utilised the degree pair $(5,1)$. To compete with this we may expect to need a degree pair such as $(3,3)$ or $(2,4)$. Theoretically it may be possible [76] to extend the quadratics method to higher degrees but there are practical difficulties and there are no known methods for finding degree pairs such as $(2,4)$ at the time of writing.

2.1.2 Sieving

Relations are found using a sieving process. There are three types of sieve: classical, lattice and line. Of these, classical sieving is rarely used as it is less efficient. Both lattice and line sieving techniques are in general use, and for a single factorisation more than one method may be employed.

We will have good reason to need to use the classical sieve later and so we include a description here.

Fix b, we define a range $[a_{\min}, a_{\max}]$ which is determined empirically. There are two sieves, one for each number field. We test the $F_i(a,b)$, $a \in [a_{\min}, a_{\max}]$ for B_i-smoothness by a sieve over the a interval using the fact that $a - b\alpha_i \in \mathfrak{p} \Leftrightarrow a \equiv br \bmod p$, where (p,r) corresponds to the first degree prime ideal $\mathfrak{p}$. The pairs that pass these tests are likely to be those we are looking for, and are subjected to gcd and trial division tests.

For each consecutive b and each number field we complete the following steps:

1. Initialise the sieve locations S_a to an approximation to $\log|F_i(a,b)|$ for $a_{\min} \leq a \leq a_{\max}$.
2. For each first degree prime ideal (p,r) in the factor base for which we have $a \equiv br \bmod p$ we subtract a low precision approximation to $\log p$ from the sieve location S_a. If, in addition, p divides both the leading coefficient of F_i and b we subtract a low precision approximation to $\log p$ from the sieve location.

3. For each number field and each a we check whether S_a is close to 0 in which case we have a report. Finally, if $\gcd(a,b) = 1$ we have a report. If we have three positive reports, that is, both $F_i(a,b)$ are thought to be smooth and $\gcd(a,b) = 1$, then we trial divide the $F_i(a,b)$, to ensure that both numbers are smooth.

We note that if the current b is even it is less time consuming to simply ignore the even a values [85]. It is also usual not to sieve over the small primes; instead replacing them with small powers. In practice we will initialise the sieve locations in such a way so as to take into account the use of small prime powers and approximated logarithms.

The lattice sieve [81] is as follows. We fix a set P_S of special prime ideals of K_1, such that for each $(q,s)_1 \in P_S$, $s \in \mathcal{R}_1(q)$ (f_1 has at least one root modulo q). For each prime ideal $(q,s)_1$ we find pairs (a,b) for which $F_1(a,b)/q$ and $F_2(a,b)$ are smooth.

1. Choose a region R of the (a,b)-plane to be sieved.
2. Choose a prime ideal $(q,s)_1 \in P_S$ and sieve only those pairs $(a,b) \in R$ for which $a \equiv bs \bmod q$.
3. We then sieve the numbers $F_1(a,b)$ with prime ideals $(p,r)_1$, $r \in \mathcal{R}_1(p)$ with $p < q$ only. We sieve the numbers $F_2(a,b)$ with the whole factor base $\mathcal{F}(B_2)$.

If a prime ideal $(q,s)_1$ does not have norm q too small a prime, then knowing $q|F_1(a,b)$ renders it more likely that $F_1(a,b)$ is smooth. We miss some smooth values of $F_1(a,b)$ that don't have divisor q, but gain in efficiency because it is quick to identify the pairs (a,b) for which $a \equiv bs \bmod q$. The pairs form a lattice in the (a,b)-plane and hence by using a reduced lattice basis can be readily identified. Sieving in step three above can be achieved in two different ways: by rows or by vectors. Let (g,h) the coordinate system with respect to the reduced basis.

- *Sieving by rows:* Fix h. Sieve over the factor base elements (p,r), $p < q$ in a similar manner to the classical sieve. This can be inefficient for larger primes.

- *Sieving by vectors:* We use the fact that the points to be sieved over form a lattice in the (g, h)-plane: a reduced basis can be formed, then the lattice generated. However such a basis may not be well-defined, in which case we are unable to sieve by vectors.

Line sieving is similar to lattice sieving by rows, but with a fixed b. We fix $(q, s) \in P_S$ then fix b; we then lattice sieve on all (a, b) for which $a \equiv bs \bmod q$; then we increment b. Incrementing b is expensive and so the number of times it is incremented is minimised.

The lattice sieve is the most advanced sieving method but it is often used in conjunction with a line siever when using a variety of different computers in parallel to sieve. It is thought that this is more efficient than using a single sieving technique. Using both techniques will lead to many duplicate relations; however, the lattice sieve alone produces duplicates, so we must filter the relations regardless.

The lattice sieve deteriorates as q increases and the line sieve as b increases. We wish to make full use of the most fertile ranges in both sieves. More practically, we wish to use both techniques as lattice sievers may be run on smaller machines but produce duplicates, while line sievers require more memory and produce no duplicates. Since we will typically sieve in parallel on a variety of machines, we can make the best use of the available resources by using both sieves.

2.1.3 Filtering and linear algebra

The aim of filtering is to reduce the amount of raw data. The linear algebra step is the practical bottleneck in NFS; the system is huge and sparse and finding a solution is a costly procedure that cannot be efficiently distributed. Therefore we wish to minimise the size and density of the final matrix while maximising the amount of information contained in the system. We note three particular parts of filtering:

1. The lattice sieve produces duplicate relations, and when used together with the line sieve a significant percentage of relations collected can be duplicates.

These relations add greatly to the size of the system and provide no new information. We remove all duplicate relations.

2. Where some prime ideal occurs exactly once for one polynomial we remove the relation that contains that prime ideal and do not add in any free relations that involve that prime ideal. Such relations can never be part of any solution set.

3. There are various ways of merging relations. The aim is to reduce the size of the matrix by combining relations. This comes at the price of increasing the density of the matrix. We wish to merge in such a way that we minimise the amount of fill in. Such methods are described and analysed in [13].

Each relation vector reduced modulo 2 forms one column of the large and sparse matrix. There are many choices for how to proceed in finding the dependencies within the system.

Algorithms for solving systems of linear equations include standard Gaussian elimination, structured Gaussian elimination [5], Lanzcos based methods [57] and Wiedemann based methods [52].

Gaussian elimination on a sparse matrix will lead to rapid fill-in as those positions that were originally zero become non-zero. Ideally we would like the matrix to remain as sparse as possible, since matrices that arise from integer factorisation are usually huge and storing them in some sparse representation is the only possibility. Structured Gaussian elimination uses certain structures which are found in matrices that arise from integer factorisation to avoid rapid fill-in. The matrix has known heavier regions as these correspond to the first degree prime ideals of small norm.

The other methods preserve the sparsity of the matrix and also utilise the fact that sparse matrices may be multiplied by vectors much faster than is usually the case. However they are still inefficient when used over GF(2) as they work with single bits. In addition, it is not possible to apply the standard Lanzcos method when working over GF(2) as approximately half the time the method will terminate with a failure condition. This can be solved by working in $GF(2^r)$ as explained in [24].

Coppersmith presented the first block Lanzcos algorithm [27] which was seen as highly complicated and hard to program [24]. This was followed shortly after by a block Wiedemann algorithm [28]. Montgomery produced his own block Lanzcos method [71] and this was used to solve some matrices produced by the number field sieve including the examples in [49].

Coppersmith's block Wiedemann [28] and Montgomery's [71] block Lanzcos methods are compared by Penninga [79] and the block Lanzcos algorithm was found to perform better. It is also the algorithm used to solve most modern large factorisation matrices. A speed up of Coppersmith's block Wiedemann algorithm (presented in [89]) is thought not to be useful when working modulo two.

2.1.4 Extraction of square roots

We need to find the square root of an algebraic integer with large coefficients. The numbers involved are of a huge size and this stage would threaten to dominate the running time of GNFS. Various methods have been presented including one in the original GNFS paper [12]. A more practical method that works only in the case d odd is found by Couveignes [29]. More recently we have a method of Montgomery, which is described and implemented in [49] with a more in depth exposition in a technical report [73]. Further work has been done on this method by Nguyen [78].

Montgomery's algorithm takes a square $\beta^2 \in \mathbb{Z}[\alpha]$ with a known factorisation into prime ideals such that each prime ideal has an even exponent and calculates β using an iterative process which utilises our knowledge of the prime ideal factorisations. The algorithm has proven to be both practical and efficient. Just as important is the fact that we are able to remove the requirement that the degree of the number field be odd. It is possible to view this problem as essentially solved.

2.1.5 Summary of the status of the main steps

Two of the four steps involved in the number field sieve are currently more easily achieved. While sieving is the asymptotic bottleneck it is eminently distributable; also the methods for sieving have been well researched. Due to Montgomery we are also able to extract square roots. The true practical bottleneck of the number field sieve is now the matrix step. Improving both the linear algebra methods available to us and the filtering of relations prior to that is necessary if we are to reduce both the computing time and space required by this step.

The choice of polynomials in the first step of the algorithm has been shown to be extremely important and is perhaps the least well understood problem. In the next chapter we will see how recent research has shown that alterations in the method of polynomial selection can improve the runtime of the whole algorithm and reduce the size of matrix we must deal with in the linear algebra step.

2.2 Smooth integers and heuristic runtime analysis

An analysis of the general number field sieve was presented in [12] and we guide the reader through this.

Definition 3

- *For* $x \geq 1$, $B \geq 1$ *let* $\Psi(x, B)$ *denote the number of* B*-smooth positive integers up to* x*.*
- $L_x[u, v] := \exp\left(v(\log x)^u(\log\log x)^{1-u}\right)$ *where* $x, u, v \in \mathbb{R}$ *and* $x > e$*. This function will be used to express the conjectured runtime. It is usual to abbreviate* $L_x[u, v + o(1)]$ *to* $L_x[u, v]$ *and note that the* $o(1)$ *is for* $x \to \infty$*.*

In chapter 1 we introduced the expression $\Psi(x, B)/x$ — the probability that a *randomly selected positive integer up to* x is B-smooth. If we wished to find B B-smooth numbers we would therefore expect to test $xB/\Psi(x, B)$ numbers. We

will work with a similar expression to this and aim to find a value of B that minimises the expectation. The expression that we will work with is slightly more general. The following theorem is the basis of the analysis of a variety of factoring algorithms and it provides a route to finding a value of B that minimises a measure of the expected number of draws to choose B numbers that are B-smooth and not greater than x. The theorem is proved in [12] for instance.

Theorem 1 *Suppose g is a function defined for all $B \geq 2$ that satisfies $g(B) \geq 1$ and $g(B) = B^{1+o(1)}$ for $B \to \infty$. Then as $x \to \infty$*

$$\frac{xg(B)}{\Psi(x,B)} \geq L_x[1/2, \sqrt{2} + o(1)]$$

uniformly for all $B \geq 2$. In addition,

$$\frac{xg(B)}{\Psi(x,B)} = L_x[1/2, \sqrt{2} + o(1)]$$

for $x \to \infty$ if and only if $B = L_x[1/2, \sqrt{2}/2 + o(1)]$ for $x \to \infty$.

Now, following [12], suppose that a factoring algorithm factoring n produces "auxiliary" numbers (e.g. the $F_i(a,b)$ in NFS) which are bounded by $x = x(n)$ and that we need to find $B^{(1+o(1))}$ of these numbers which are B-smooth, for $n \to \infty$. If we assume that the auxiliary numbers are *just as likely to be B-smooth as random integers up to x* then, like above, we would expect to test $xB^{(1+o(1))}/\Psi(x,B)$ numbers for B-smoothness. If we further assume that the time to test one of the numbers for B-smoothness is $B^{o(1)}$ we obtain an expected runtime of

$$\frac{xB^{(1+o(1))}}{\Psi(x,B)}$$

to find the required B-smooth numbers. We are then able to utilise the above theorem to minimise the runtime of this stage producing $B^{(2+o(1))} = L_x[1/2, \sqrt{2} + o(1)]$. If all other steps in our given algorithm take at most this runtime then that is the complete runtime of the algorithm.

In order to complete the analysis we need to estimate the size of the auxiliary numbers that are encountered in the factoring algorithm — that is, we require an estimate of the size of the numbers that we wish to be smooth in terms of the size of the number we are factoring, n.

We noted in the introduction that in the quadratic sieve the numbers we test for smoothness are of size approximately $x = O(n^{1/2})$, hence we have a heuristic asymptotic runtime for the quadratic sieve of:

$$\begin{aligned} L_x[1/2, \sqrt{2}] &\approx \exp\left((\sqrt{2} + o(1))(\log n^{1/2})^{1/2}(\log\log n^{1/2})^{1-1/2}\right) \\ &\approx \exp\left((1 + o(1))(\log n)^{1/2}(\log\log n)^{1/2}\right) \\ &= L_n[1/2, 1] \end{aligned}$$

For some factoring algorithms this argument can give a rigorous complexity analysis, however for others we must make various heuristic assumptions. In particular we have assumed that the auxiliary numbers are just as likely to be B-smooth as random integers of the same size — matters are in fact not that simplistic (as will we see in subsequent chapters).

As previously noted it is the small size of the numbers we test for smoothness that gives the number field sieve an advantage over other algorithms in its family. In fact the number field sieve is the first factoring algorithm to have the size of x bounded by a term that is subexponential in the size of n and hence the first that can achieve better than $u = 1/2$ in $L_n[u, v]$.

Runtime of GNFS

Following [12] we must bound the auxiliary numbers; that is, those numbers generated in the sieving step that we wish to test for smoothness. We assume that the sieve parameters $|a|$ and b are bounded by u, that we have one linear polynomial and one non-linear polynomial of degree d, the latter having being produced by the base-m method. At each sieve location we assess the integer $F_1(a, b)F_2(a, b)$ which has absolute bound:

$$\begin{aligned} |F_1(a, b)F_2(a, b)| &\leq (d + 1)m^2u^{d+1} \\ &\leq 2dm^2u^{d+1} \\ &\leq 2dn^{2/d}u^{d+1} \end{aligned}$$

The coefficients of F_1 are bounded by m and we have $m \leq n^{1/d}$. Hence the number $x = 2dn^{2/d}u^{d+1}$ is a bound on the numbers that we test for smoothness.

Under the assumption that the values are as likely to be smooth as randomly selected integers of the same size, using the above bound and the theorem of the latter section it is shown in [12] that with optimal choices of B and u the asymptotic runtime is

$$\exp\left((1+o(1))\left(d\log d+\sqrt{(d\log d)^2+4\log n^{1/d}\log\log n^{1/d}}\right)\right)$$

with a bound on x of $L_n[2/3,(64/3)^{1/3}]$.

Further to this it is noted that the minimum value must occur when $(d\log d)^2$ and $(\log n^{1/d}\log\log n^{1/d})$ are of the same magnitude. That is,

$$(d\log d)^2=\mathcal{O}\Big(\log n^{1/d}\log\log n^{1/d}\Big).$$

In [12] it is noted that this occurs when $d=C(\log n)^{1/3}(\log\log n)^{-1/3}$ and that optimising produces $C=(3^{1/3}+o(1))$ for $n\to\infty$. This value of d, will then produce the heuristic asymptotic runtime of

$$L_n[1/3,(64/9)^{1/3}].$$

This runtime is achieved because the parameter choices for u and d can force the numbers that we test for smoothness to have subexponential size when compared with n.

2.3 Large prime variants

The key idea of the large prime variants is that we relax the requirement for the algebraic numbers $a-b\alpha_i$ to be B_i-smooth and instead allow *up to* j_i large primes. Hence we find relations by finding pairs (a,b), a coprime to b, such that $a-b\alpha_i$ is divisible only by primes in the factor base and up to j_i primes with norm greater than the factor base bound but smaller than a large prime bound.

We call such a relation a j_1,j_2-partial relation (and refer to relations with $j_i=0$ as full relations). Current factorisations use $j_1,j_2\leq 3$ and even this is still unusual, more usually one of the j_i would be capped at 2.

The collection of possible large prime variants cannot really be separated from the main algorithm. In fact even the original special case factorisations used one large prime. However the inclusion of large primes in the main description would have confused matters unnecessarily.

We need to explain the required alterations to the algorithm and also underline why this variation is so important in practice. This is simpler in the 1, 1-partial case so we will consider that first.

2.3.1 Up to one large prime on each side

These relations are found by making a simple alteration to the sieving process. In addition, they require very little extra processing to be useful and can be found in large quantities. Despite the fact that using these large prime relations provides no asymptotic speed up it is invaluable in practice.

As explained earlier, after sieving we receive reports if the sieve location is "close" to 0. If we allow a greater tolerance when checking "closeness" we get more reports. Then when trial dividing in order to confirm or deny B-smoothness, we can also keep any relation that (after trial division by the factor base primes) leaves a remainder less than B_i^2 which must be prime.

We produce a set of full relations and partial relations with one large prime on either or both sides. During filtering, any relations that contain a prime not found in any other relation (large or otherwise) are removed; hence any large prime that can never be contained in the set S creates very little overhead.

The main negative side effect is that the increase in size of the factor base necessitates an increase in the number of relations required to form a useful matrix. In the past, the partial relations would have been specifically merged with each other in order to remove all the large primes; however this creates a denser matrix. Advances in merging and filtering have led to a more subtle approach where relations are merged in such a way as to attempt to minimise growth in matrix density. This will not necessarily absorb all of the large primes — as other primes are removed in their place if this is more advantageous.

2.3.2 Additional large primes

Large prime relations with two or more large primes on either side (or both sides) are not so easily distinguished. In this case, given an auxiliary number has triggered a report, the remainder after dividing out all the factor base elements needs to be factored to find the large primes — this increases the overhead. In addition, if a reported pair (a, b) expected to have say, two large primes in fact has only one this prime is usually too large and it is highly unlikely to turn up in a second relation. Hence, despite the fact that this method can produce an even larger quantity of relations for a reasonable overhead, we must be aware of the trade off — especially as we increase the quantity of large primes permitted. In addition, the fact that the relations involve more large primes means that they will lead to either a larger matrix or a denser matrix than would otherwise have been the case. It is for these reasons that the number of large primes generally used is two on each side or two on one side and three on the other.

For a more in depth description of the issues and discussion of the trade off Cavallar [14, 20] compares using three large primes on each side with two. Other authors [38, 64] provide additional background in this area.

2.4 Multiple polynomial number field sieve

Another possibility is that of using more than two polynomials. This was first suggested by Coppersmith [26]. A more practical algorithm is given by Huizing [40, 41] where the quadratics method is used to create multiple quadratic polynomials.

Huizing's method

Take k polynomials $f_i(X) = \sum_{j=0}^{d_i} a_{i,j} X^j \in \mathbb{Z}[X]$, $i = 1, \ldots, k$ such that

1. $f_1(m) \equiv f_2(m) \equiv \ldots \equiv f_k(m) \equiv 0 \bmod n$.
2. The f_i are irreducible $\forall\, i = 1, \ldots, k$.

3. $\text{cont}(f_i) = 1$, $\forall\, i = 1, \ldots, k$.

4. $f_i \neq f_j$, $\forall\, i \neq j$.

Let $\alpha_i \in \mathbb{C}$ be such that $f_i(\alpha_i) = 0$. Let $\mathbb{Q}_n$ be the ring of rational numbers with denominator coprime to n and note the natural homomorphisms $\varphi_i : \mathbb{Q}_n \to \mathbb{Z}/n\mathbb{Z}$ determined by $\varphi_i(\alpha_i) \equiv m \bmod n$.

We sieve in a similar manner to usual but we say that a coprime pair (a, b) is a (j_1, j_2)-relation if $1 \leq j_1 < j_2 \leq k$ and both integers $F_{j_1}(a, b)$ and $F_{j_2}(a, b)$ are smooth. We will write this $(a, b)_{j_1, j_2}$. Note that for a pair (a, b) with $t \geq 2$ smooth integers $F_{j_1}(a, b) \ldots F_{j_t}(a, b)$ we can make $t - 1$ relations.

The matrix is formed as usual with one column for each relation and a row for each factor base element, however we have k factor bases. This results in a matrix that has a differing structure.

Huizing implemented this method using multiple degree 2 polynomials since the method for choosing two degree 2 polynomials is easily extended. It is not known how one would go about producing multiple polynomials of other degrees.

Huizing concluded that despite a speed up in classical sieving there was no positive effect on line sieving and as a result this method is not currently considered to be practical.

2.5 Summary

We have described the general number field sieve, excepting the most modern polynomial selection methods which will be considered in the succeeding chapter. We have outlined the asymptotic complexity argument that provides us with a heuristic runtime.

Finally we have described the large prime variants which are of huge practical importance and noted the main idea behind the multiple polynomial sieve variants which have yet to be proved useful.

Chapter 3

Background: Yield and polynomial selection

Steps have been made which increase our understanding of the yield of the number field sieve and which are vital to improving the practical performance of the algorithm. In particular we are interested in estimating and maximising the quantity of relations produced by the sieving step.

Used alongside sieving experiments estimates of sieving yield can be beneficial in selecting appropriate parameters — of particular concern when the sieving process is to be a lengthy one or when outside participants are being asked to provide processor time. In addition, the method of estimating can be used to consider possible variants of the number field sieve that will not impact on the asymptotic behaviour but which may produce a practical speed up. Cavallar used estimates alongside sieve tests when considering the viability of the three large prime variant of NFS and this may also prove useful when considering NFS variants which employ multiple number fields.

In the area of polynomial selection knowledge of the main criteria that affect yield and how yield can be estimated has led to impressive improvements. Murphy [75, 76] has used methods that rank a selection of polynomials (without performing sieve tests) based on a raw estimate of the quantity of relations likely to be produced with certain parametrisations. This can then be used to select better polynomials and associated parameters for the general number field sieve. These

methods have been further improved by Gower [46].

Key to the estimation of yield is an understanding of the quantity of smooth integers below a bound x or in an interval $[x_1, x_2]$. There are a number of particularly pertinent results which we will summarise before considering how these have been used in the case of the number field sieve.

3.1 Smooth and semismooth integers

De Bruijn's [34] function $\Psi(x, y)$, introduced earlier, denotes the number of positive integers up to x that are y-smooth. A widely accepted method of approximating this function makes use of the Dickman rho function, for $x \in \mathbb{R}$, $x \geq 0$ Dickman's ρ function is defined by:

$$\begin{aligned} \rho(x) &= 1 \text{ for } 0 \leq x \leq 1 \\ x\rho'(x) + \rho(x-1) &= 0 \text{ for } x \geq 1 \end{aligned}$$

This function is piecewise analytic and agrees with the analytic function ρ_k on the interval $[k-1, k]$, $k \geq 1$ where for $0 \leq \xi \leq 1$

$$\rho_k(k - \xi) = \sum_{i=0}^{\infty} c_i^{(k)} \xi^k \text{ for } k = 1, 2, \ldots.$$

Bach and Peralta [4] give an efficient method of calculating the coefficients (the method is that of Patterson and Rumsey) they give

$$\begin{aligned} c_0^{(1)} &= 1, & c_0^{(2)} &= 1 - \log 2, \\ c_i^{(1)} &= 0, & c_i^{(2)} &= 1/(i2^i) \text{ for } i \geq 1, \\ c_i^{(k)} &= \sum_{j=0}^{i-1} \frac{c_j^{(k-1)}}{(ik^{i-j})}, & c_0^{(k)} &= \frac{1}{(k-1)} \sum_{j=1}^{\infty} \frac{c_j^{(k)}}{(j+1)} \text{ for } k > 2. \end{aligned}$$

It is noted that to compute ρ to IEEE standard double precision 55 coefficients should be calculated. There are more suitable methods for approximating ρ to very high precision but these are not necessary in this circumstance.

Another method which requires greater precision is due to Marsaglia, Zaman and Marsaglia [67]; Cavallar [20] successfully combines both methods to reduce the time required and increase the working precision.

Dickman found the earliest approximation for $\Psi(x, y)$, if we set $\alpha = (\log y)/(\log x)$ then we have

$$\Psi(x, x^\alpha) \approx x\rho\left(\frac{1}{\alpha}\right).$$

De Bruijn improved on this and Bach and Peralta [4] note that his results imply that for $0 < \gamma \leq \alpha < 1$, $x^\gamma \geq 2$

$$\Psi(x, x^\alpha) = x\rho\left(\frac{1}{\alpha}\right) + O\left(\frac{x}{\gamma \log x}\right).$$

and hence we define the first function we will use in finding estimates of the yield of full relations:

$$G_0(\alpha) := \lim_{x\to\infty} \frac{\Psi(x, x^\alpha)}{x} = \rho\left(\frac{1}{\alpha}\right).$$

Since we would also like to estimate the large prime relation yield we must go further and produce approximations for the quantity of i-semismooth integers. Bach and Peralta extended De Bruijn's function to $\Psi(x, y, z)$ which denotes the number of positive integers up to x which are z-smooth possibly excepting one prime divisor which is less than y, $z < y$. We will instead work with the slightly different $\Psi_1(x, y, z)$. For reasons that will become clear we will define $\Psi_i(x, y, z)$ to be the number of positive integers up to x which are z-smooth except for *exactly* i prime divisors *greater than* z but less than y.

Setting $\beta = (\log z)/(\log x)$ Bach and Peralta proved for $0 < \alpha < \beta < 1$

$$G_1(\alpha, \beta) := \lim_{x\to\infty} \frac{\Psi_1(x, x^\beta, x^\alpha)}{x} = \int_\alpha^\beta \rho\left(\frac{1-\lambda}{\alpha}\right)\frac{d\lambda}{\lambda}$$

which is needed to form an approximation for the yield of one large prime relations. Lambert [56] continued this work proving for $0 < \alpha < \beta < 1/2$

$$G_2(\alpha, \beta) := \lim_{x\to\infty} \frac{\Psi_2(x, x^\beta, x^\alpha)}{x} = \frac{1}{2}\int_\alpha^\beta \int_\alpha^\beta \rho\left(\frac{1-\lambda_1-\lambda_2}{\alpha}\right)\frac{d\lambda_1}{\lambda_1}\frac{d\lambda_2}{\lambda_2}.$$

as required in the case of two large primes. Cavallar [14, 20] and Zhang [91]

(independently) completed the generalisation proving for $0 < \alpha < \beta < 1/i$

$$\begin{aligned} G_i(\alpha,\beta) &:= \lim_{x\to\infty} \frac{\Psi_i(x,x^\beta,x^\alpha)}{x} \\ &= \frac{1}{i!}\int_\alpha^\beta \int_\alpha^\beta \cdots \int_\alpha^\beta \rho\left(\frac{1-\lambda_1-\lambda_2-\cdots-\lambda_i}{\alpha}\right)\frac{d\lambda_1}{\lambda_1}\frac{d\lambda_2}{\lambda_2}\cdots\frac{d\lambda_i}{\lambda_i}. \end{aligned}$$

It is by use of the functions G_i, $i = 0, 1, 2, \ldots$ that we are able to estimate the yield of the number field sieve and other similar algorithms. Cavallar provides, in section 2.6 of [20], an analysis of how well the G_i approximate the $\Psi_i(x, x^\beta, x^\alpha)/x$ *under the assumption* that $\rho(1/\alpha)$ is a good approximation for $\Psi(x, x^\alpha)/x$. Hunter and Sorenson consider the latter question in [51, Table 2].

The G_i variations are based on the lower estimate $x\rho(1/\alpha)$ and these are not the most sophisticated variations, instead of using G_0 we can instead use the function

$$H_0(x,y) := \rho\left(\frac{\log x}{\log y}\right) + \frac{1-\gamma}{\log x}\rho\left(\frac{\log x}{\log y} - 1\right).$$

Corresponding approximations can be defined for $i > 0$ (see [20] for details). These more sophisticated approximations were utilised by Boender [7] and Murphy [76]; more specifically they used an interval form of the latter function to estimate the quantity of full relations found on an interval $[x_1, x_2]$ rather than simply less than x. In order to do this they used a formula due to Hildeband and Tennenbaum [48]. However, the equivalent calculations for the semismooth integers are extremely time consuming and there is no obvious way to generalise such a formula to a sieve region with $b > 1$.

Approximations of the quantity of smooth numbers up to x or in an interval are not immediately useful in the case of the number field sieve since we are considering polynomial values taken over a sieve region. Murphy did use knowledge of the above approximations to produce a ranking within a set of polynomial pairs and for this the methods used were suitable.

All of the above apply only to random numbers — not to the integral values take on a polynomial $F(X, Y)$ over a region $R = [a_1, a_2] \times [1, b)$. The introduction of a measure of how much the probability of the polynomial values being smooth differs from the probability of random integers of the same size being smooth

allows us to utilise the estimates.

In the next section we will talk about a function $\alpha(F)$, described in detail by Murphy [76] which captures this key measure. Hence we see the polynomial values x as having the same probability of smoothness as randomly selected integers of size $xe^{\alpha(F)}$.

3.2 Properties that affect polynomial yield

We require polynomials that will produce many smooth values. More specifically [40, 49, 76]:

1. The maximum values of $|F_i(a,b)|$ should be small to increase the likelihood that the values taken by F_1 and F_2 will be smooth over the primes below B_1 and B_2 respectively.
2. Polynomials with a real root close to $\frac{\max(|a|)}{\max(|b|)}$ are a good choice since this will also increase the likelihood of values being smooth.
3. Polynomials which have many roots modulo small primes have a higher probability of taking values that are small after dividing by these small primes. This again increases the likelihood of finding smooth values.
4. Polynomials with small Galois group size are preferable as this maximises the density of free relations.

These can be separated into size and root properties.

The effect of size on the yield is clear, and the probability of a randomly selected number x of a fixed size being smooth is well understood. Stated simply, given n to be factored and the degree d of the non-linear polynomial we must select a pair of polynomials F_1, F_2 which have small size over the sieving region. However, d is the first parameter that must be chosen. As described in the preceding chapter, we have a way of selecting d that minimises the asymptotic runtime as $d \to \infty$ however, d grows very slowly with n and hence in practice we may need to take other factors into consideration. Murphy presents a table of best d values for

general integers in the current range of interest [76, section 3]. At the current time we would expect to use $d = 5, 6$ for most factorisations undertaken.

The effect of the root properties is perhaps less clear. The key idea is that the numbers that we wish to be smooth are not in fact randomly selected integers but are numbers of a specific form. Montgomery proposed a way to compare the smoothness probabilities of numbers of this form with smoothness probabilities of random numbers. This was originally suggested in connection with the quadratic sieve and was used to select parameters that increased the likelihood that values of the quadratic polynomial would be smooth [7]. Later, similar ideas were used to great effect by Murphy to improve estimates of polynomial yield and hence polynomial selection in the general number field sieve.

Let v be some value in our sieve region. If we were working with a perfect sieve then the full contribution of each prime in the factor base would be removed from each value v we sieve over. In fact we start with the logarithm of the value v and subtract a low precision approximation to the logarithm of each contribution. Following Murphy [76] we summarise the derivation of the alpha function $\alpha(F)$.

Derivation of $\alpha(F)$

Definition 4 *Denote by* $\mathrm{ord}_p(v)$ *the exponent of the largest prime power of* p *dividing* v. *Then* $\mathrm{cont}_p(v)$ *is the expected value of* $\mathrm{ord}_p(v)$ *as* v *ranges across some sample* R.

So after sieving we would have

$$\log v - \sum_{p \leq B} \mathrm{cont}_p(v) \log p.$$

We are interested in the difference between this value when $v = F(a, b)$ and when v is a randomly selected integer of the same size.

We will compute the expected contribution of a prime $p < B$ to $F(a, b)$. That is, we calculate the average exponent of p in the factorisation of $F(a, b)$, this is denoted $\mathrm{cont}_p(F)$. In addition we compute the equivalent value for random

numbers and denote this $\text{cont}_p(r)$.

For a random value r we expect that powers p^k for an integer $k > 1$ may also divide r. Therefore we expect that the average contribution of p to r is

$$\text{cont}_p(r) = \frac{1}{p} + \frac{1}{p^2} + \cdots = \frac{1}{p-1}.$$

We now consider polynomial values of the form $F(a, b)$ with $\gcd(a, b) = 1$.

For p an unramified prime (and hence a prime which does not divide the discriminant of f [21]) the contribution of p to f is from a single root of $f \bmod p$. Such primes were termed *well behaved* by Murphy. If a prime is not well behaved we cannot use the following method but must either compute the prime decomposition of $\langle p \rangle$ or, if factoring the discriminant is too expensive, use a sampling method to estimate the correct value for all small primes (which have a greater impact on the yield).

Let p be a well behaved prime. We will first consider values of the form $f(a) = F(a, 1)$ since this is easier than the general case (and will be of interest when line sieving). Let q_p be the number of distinct roots of $f(X) \bmod p$. By Hensel lifting each root corresponds to a distinct root modulo p^k, $k > 1$ and for each root modulo p we have a distinct probability of being divisible by p hence for *each root* we have the contribution $1/(p-1)$ producing

$$\text{cont}_p(f) = \frac{q_p}{p-1}.$$

Matters are somewhat more complicated when we work with polynomial values of the form $F(a, b)$, a, b coprime as we no longer have a unique correspondence between the roots of f modulo p and the roots modulo p^k for $k > 1$. We also have the possibility that $p|b$, since $F(X, Y) = Y^d f(X/Y)$ if p also divides the leading term of f we have $p|F(a, b)$. Murphy termed such roots projective roots.

Let q_p now be the number of roots modulo p of $F(X, Y)$, this set would include both the roots of $f \bmod p$ and any projective roots.

The full contribution of p to the value $F(a,b)$ with a, b coprime is given by

$$\mathrm{cont}_p(F) = \frac{q_p p}{p^2-1}.$$

Following an argument of Murphy [76, section 3.2.2] we see this by considering the probability that p^k divides $F(a,b)$, a,b coprime and then summing over k. Since F is a homogeneous polynomial we may view the coprime pairs (a,b) as points on the projective line. We consider a root $X/Y \bmod p^k$ of F, there are 3 cases:

1. $X/Y \equiv s \bmod p^k$, $s \not\equiv 0 \bmod p$, for some $s \in \mathbb{Z}/p^k\mathbb{Z}$. There are $\phi(p^k)$ possible such s and for each such s given an $X \in \mathbb{Z}/p^k\mathbb{Z}$, $X \not\equiv 0 \bmod p$, Y is uniquely determined. There are $\phi(p^k)$ possible such X and hence $\phi(p^k)$ possible pairs (X,Y) for each s.

2. $X/Y \equiv s \bmod p^k$, $s \equiv 0 \bmod p$. Clearly there are only p^{k-1} such s. Setting $X \equiv 0 \bmod p$ we see that there are then $\phi(p^k)$ possible invertible $Y \in \mathbb{Z}/p^k\mathbb{Z}$ and hence $\phi(p^k)$ possible pairs (X,Y) for each s.

3. $Y/X \equiv s \bmod p^k$, $s \equiv 0 \bmod p$. In this case we write $X/Y = \infty$ and refer to these as the projective roots. We are in a similar situation to case 2 (exchange values of X and Y).

Finally we note that $\phi(p^k) = p^{k-1}(p-1)$ and that there are

$$\phi(p^k) + 2p^{k-1} = p^{k-1}(p-1+2) = p^{k-1}(p+1)$$

possible classes. Hence a coprime pair $(X,Y) \in \mathbb{Z}/p^k\mathbb{Z} \times \mathbb{Z}/p^k\mathbb{Z}$ selected uniformly at random will be in any one of the 3 cases with probability $1/p^{k-1}(p+1)$. We know that we have q_p distinct roots of $F(X,Y)$ modulo p so the probability that p^k contributes to any pair is $q_p/p^{k-1}(p+1)$, then $1/p$ of these would be counted a second time when we consider p^{k+1} so the contribution from p^k is thus:

$$\frac{q_p}{p^{k-1}(p+1)}\left(1-\frac{1}{p}\right).$$

Logarithmically p^k contributes k appearances of p and hence:

$$\begin{aligned}
\mathrm{cont}_p(F) &= \sum_{k=1}^{\infty} \frac{kq_p}{p^{k-1}(p+1)} \left(1 - \frac{1}{p}\right) \\
&= \frac{q_p}{p+1} \left(1 - \frac{1}{p}\right) \sum_{k=1}^{\infty} \frac{k}{p^{k-1}} \\
&= \left(1 - \frac{1}{p}\right)^{-1} \frac{q_p}{p+1} \\
&= \frac{q_p p}{p^2 - 1}.
\end{aligned}$$

Murphy provides computational support for these heuristics.

Our aim was to summarise the technology required to enable the comparison of the probability of smoothness of the auxiliary numbers arising in NFS and random integers. As noted above after sieving we would obtain

$$\log v - \sum_{p \leq B} \mathrm{cont}_p(v) \log p.$$

For random values r this produces

$$\log r - \sum_{p \leq B} \frac{1}{p-1} \log p,$$

now setting either $v = f(X)$ or $v = F(X, Y)$ and taking the difference we define

$$\alpha(v) = \sum_{p \leq B} \left(\frac{1}{p-1} - \mathrm{cont}_p(v) \right) \log p$$

with the appropriate value of $\mathrm{cont}_p(v)$ from above. Hence we have $\log v = \log r + \alpha(v)$ and we proceed with the assumption that the auxiliary numbers $F(a, b)$ we test for smoothness in the number field sieve behave like randomly selected integers of size $F(a, b)e^{\alpha(F)}$. Clearly we would prefer to work with polynomials F for which $\alpha(F) < 0$.

Finally, we note that the value of $\alpha(F)$ is extremely sensitive to changes in the value of q_p when p is small. This can be seen immediately from the form of $\alpha(v)$: the value of p dominates the calculation as p grows. To force $\alpha(F)$ to be more negative we would need to ensure that F has many roots modulo small primes.

3.3 Estimating yield over a sieve region

We will assume we have one linear polynomial and one of higher degree although other variants have been used [40, 77]. We will assume that $d_2 = 1$ in particular.

A number is said to be (j, B, L)-smooth if it has exactly j prime factors greater than B and less than or equal to L and all the remaining prime factors are less than or equal to B.

The sieving step then consists of finding pairs (a, b) such that a and b are coprime, $F_1(a, b)$ is (j_1, B_1, L_1)-smooth and $F_2(a, b)$ is (j_2, B_2, L_2)-smooth. We recall that such pairs are called j_1, j_2-partial relations and that if $j_1 = j_2 = 0$ the relation is referred to as a full relation. It is usual for $j_1, j_2 \leq 2$ and we will assume that this is the case (although we note that Cavallar has produced initial data regarding the use of $j_1 = 3$ [14]).

Following Cavallar [20] we will assume that the sieve region is $R = [-a, a] \times [1, b] \cap \mathbb{Z} \times \mathbb{Z}$. Although other sieve regions can be used the advantages and disadvantages of using other more complex regions are less well understood.

We assume that $\Psi(x, B)/x$ is the portion of B-smooth numbers among the numbers from 1 to x. The average size of these numbers is $(1 + x)/2$ which we note is approximately $\bar{x} = x/2$. In contrast, the average size of $F(X, Y)$ over the continuous region $R_c = [-a, a] \times [1, b]$ is

$$\bar{F} = \frac{\iint_{R_c} |F(X, Y)| dX dY}{\iint_{R_c} dX dY}.$$

This implies that we may be able to treat the values $F(a, b)$ like random values of average size $\bar{x}' = \bar{F} e^{\alpha(F)}$ that is, we would use $\Psi(x', B)/x'$ with $x' = 2\bar{x}'$ to approximate the portion of B-smooth polynomial values among the (a, b) pairs from R with $\gcd(a, b) = 1$.

In the sieving region we have approximately $Z = 6/\pi^2 \iint_{R_c} dX dY$ pairs such that $\gcd(a, b) = 1$ [54, section 4.5.2]. Hence we expect $Z \frac{\Psi(x', B)}{x'}$ B-smooth norms among them.

Cavallar then used $Z\frac{\Psi_j(x',L,B)}{x'}$ as an approximation for the number of (j,L,B)-smooth norms in the sieving region. Further to this it was assumed that the smoothness of $F_1(a,b)$ is not related to the smoothness of $F_2(a,b)$ and hence that we could calculate

$$Z\frac{\Psi_{j_1}(x_1',L_1,B_1)}{x_1'}\frac{\Psi_{j_2}(x_2',L_2,B_2)}{x_2'}$$

as an approximation of the number of j_1, j_2-partial relations.

Hence for $i = 1, 2$, with number fields K_i, defined by polynomials F_i, with factor base bounds B_i and large prime bounds L_i we let $x_i' = 2\bar{F}_i e^{\alpha(F_i)}$, $\alpha_i = \log_{x_i'} B_i$, $\beta_i = \log_{x_i'} L_i$ we calculate

$$ZG_{j_1}(\alpha_1,\beta_1)G_{j_2}(\alpha_2,\beta_2)$$

(and $ZH_{j_1}(x_1',B_1,L_1)H_{j_2}(x_2',B_2,L_2)$ should we wish) using the high precision approximations to $\rho(x)$ mentioned above in order to estimate the quantity of relations produced by the sieving process.

3.4 Polynomial selection for general integers

Two problems were looked at by Murphy: that of generating large samples of polynomials which are small and have good root properties, and that of selecting the best polynomials from these samples, with the base-m method being the underlying method of generating polynomials. The two quadratics method is not used.

Two forms of polynomial can be produced: non-skewed, in which all coefficients are as small as possible and skewed in which only some coefficients (usually a_d, a_{d-1}, a_{d-2}) are small and the coefficient size generally increases in absolute value from a_d to a_0. In conjunction with these distinct forms of non-linear polynomial different sieving regions are used. In the case of the non-skewed polynomial a standard sieving region with parameters $-u \leq a \leq u, 1 \leq b \leq u$ for some $u \in \mathbb{Z}$ is used. In the case of the skewed polynomials we use a rectangle whose length (a direction) to width ratio is greater than 1. The ratio is chosen based on the individual polynomial.

Using Murphy's ideas it is possible to generate polynomial pairs, in particular, pairs with highly skewed non-linear polynomials with excellent root properties. Both the RSA-140 and RSA-155 factorisations [17, 18] use polynomials produced in this way. In both factorisations the amount of time spent isolating a good polynomial was only a fraction of the total sieving time (the full yield is compared with a skewed pair of polynomials with average yield, data from [17, 18]):

	Poly. selection	Sieving	Full yield
RSA-140	60 MIPS years	2000 MIPS years	8 times average yield
RSA-155	100 MIPS years	8360 MIPS years	13.5 times average yield

RSA-140 took roughly half the expected time to factor if the expected time is extrapolated from the time taken to factor RSA-130 however, the search for a good polynomial was truncated due to practical reasons. RSA-155 took roughly a quarter of the time expected on extrapolation from RSA-130 and about a half of the time expected on extrapolation from RSA-140. It appears that such polynomials have yields $10 - 15$ times greater than the average selection [76, section 6]. This is significant in that it vastly reduces the number of machines required in the sieving step.

3.4.1 Finding good polynomial pairs

The base-m method of generating polynomials described in the previous chapter underlies the generation of large sets of good prospective polynomials. For fixed d we seek $m \approx n^{1/(d+1)}$ such that $f(m) \equiv 0 \bmod n$, f of degree d. We start with the standard f produced by the base-m method and adjust it so that the coefficients lie between $-m/2$ and $m/2$. Heuristically this is sensible as the coefficients will be smaller in absolute value, for a given m we will call this adjusted polynomial f_m.

Murphy provides a way to choose m and f_m with good combinations of size and root properties. In addition, when considering skewed polynomials variants of f other than just f_m are sought.

Definition 5 *An adjusted base-m polynomial f_m will be called χ-small when $|a_i|/m \leq \chi$, $\forall\ i = 1, \ldots, d$. If the value of χ is unimportant such a polynomial will be referred to as small.*

Non-skewed polynomials

Firstly, for the polynomial to be small it is necessary for a_d and a_{d-1} to be small; the former can be achieved by choosing m appropriately, if we also want a_{d-1} to be small we need to choose values of m close to where the value of a_d changes.

Secondly we would like to force f_m to have better than average root properties. This is done by forcing $F_m(X, Y) = Y^d f_m(X/Y)$ to have good projective roots modulo small primes. Non-projective roots are not controlled in any fashion.

Hence Murphy produces the method [76, procedure 5.1.4]:

1. Select suitable bounds $\chi_1, \chi_2, \chi_1 \leq |a_d|/m \leq \chi_2$ that ensure $|a_d|$ is significantly smaller than m. This will give us a range of a_d values:

$$\exp\left(\frac{d\log\chi_1 + \log n}{d+1}\right) \leq |a_d| \leq \exp\left(\frac{d\log\chi_2 + \log n}{d+1}\right)$$

   and a range of m values:

$$\exp\left(\frac{\log n + \log\chi_2}{d+1}\right) \leq m \leq \exp\left(\frac{\log n + \log\chi_1}{d+1}\right).$$

2. Choose a cofactor c of a_d to be a product of many small p^k, p prime, $k \geq 1$. For each a_d with cofactor c in the range, retain the values of m for which $|a_{d-1}|/m \leq \chi$ for some $\chi \geq \chi_2$.

3. For each m remaining calculate the other coefficients of f_m and, if these are small, an approximation of $\alpha(F)$. Retain those f_m for which all quantities are sufficiently small. Repeat from step 2 with varying values for c.

Skewed polynomials

The aim in this case is the same — to produce polynomials with unusually good characteristics, however, we relax the restrictions on size on the lower coefficients and attempt to find highly skewed polynomials with exceptionally good root properties. In order to compensate for relaxing the size restrictions on the lower order coefficients we also skew the sieve region. In this way we produce a polynomial and a sieve region that when used together ensure small polynomial values with an increased probability of being smooth.

Starting with adjusted base-m polynomials f_m there are two operations that are applied successively in order to achieve this.

1. Translation by t: $f_{m_t}(X) = f_m(X - t)$, $t \in \mathbb{Z}$, $m_t = m + t$.
2. Rotation by P: $f_{m_P}(X) = f_m(X) + P(X)(X - m)$, $P \in \mathbb{Z}[X]$ with the degree of P smaller than that of f_m.

Translation by an integer will not affect the root properties of the polynomial but can improve the size of the coefficients. It is also used to ensure that the resulting polynomials are central on the X-axis.

Rotation by a polynomial P is the key operation and it can alter both the size and root properties. Murphy uses only linear P however Gower [46] makes the necessary adjustments to enable us to utilise P of higher degree. Two different kinds of rotation are used with different aims, firstly rotations are used to produce polynomials that take particularly small values over the sieving region (the skew of the sieve region is altered each time we rotate), the second is used to produce polynomials with good root properties. The procedure is far more involved than that of the previous section [76, procedure 5.1.6]:

1. Construct a_d divisible by many small p^k, p prime, $k \geq 1$, calculate $m = \lfloor (n/a_d)^{1/d} \rfloor$. Compute the integral and non-integral parts of

$$\frac{n - a_d m}{m^{d-1}} = a_{d-1} + \frac{a_{d-2}}{m} + O(m^{-2})$$

retain those (a_d, m) for which a_{d-1} and a_{d-2} are sufficiently small (compared to m).

2. We now adjust f_m with the view to skewing it further and reducing its size over a skewed region. Let S be a rectangular region defined by $|X| < \sqrt{s}$, $|Y| < 1/\sqrt{s}$ and define

$$\begin{aligned} P(X) &= c_1 X - c_0 \\ f_{m_{P,t}}(X) &= f_m(X-t) + P(X-t)(X-t-m) \\ F_{m_{P,t}}(X,Y) &= Y^d f_{m_{P,t}}(X/Y) \end{aligned}$$

Apply a multi-variable minimisation to minimise

$$\iint_S \iint_S X, Y) dX dY$$

with respect to the variables s, c_0, c_1, t (treated as real variables). Round the outcomes for c_0, c_1, t to the nearest integer and recompute s. Finally estimate the average logarithmic size over the region:

$$I(F,S) = \log\left(\sqrt{\iint_S F^2(X,Y) dX dY}\right)$$

and retain polynomials for which this is sufficiently small.

3. Having produced small, highly skewed polynomials we now search among polynomials of these sizes, using rotations, for those with good root properties. We use a sieve-like process to identify $j_0, j_1 \in \mathbb{Z}$ (typically with $|j_1| << |j_0|$) for which

$$f_{j_1,j_0}(X) = f_{m_{P,t}}(X) + (j_1 X - j_0)(X - m).$$

Fix $j_1, p^k, k \geq 1$, p a small prime. Use a finite difference method to rapidly compute $f_{j_1,j_0}(l) \bmod p^k$, $l = 0, \ldots, p^k - 1$. For each l solve a linear congruence to find $j_0 \in \mathbb{Z}/p^k\mathbb{Z}$ for which $f_{j_1,j_0}(l) \equiv 0 \bmod p^k$ and then estimate $\text{cont}_{p^k}(F_{j_1,j_0})$ recording it in an array of length p^k in the position corresponding to j_0. Record $\text{cont}_{p^k}(F_{j_1,j_0})$ at any projective roots. Once we have completed this modulo p^k replicate the array throughout the space. Repeat for all small p and j_1. This will result in a (j_1, j_0)-array where each position approximates the value of $\alpha(F_{j_1,j_0})$ (using the primes considered).

4. The average size $I(F_{j_1,j_0}) \approx I(F_{m_{P,t}})$ so we give each F_{j_1,j_0} a rating of

$$I(F_{j_1,j_0}) + \alpha(F_{j_1,j_0}),$$

if this is sufficiently small then the coefficients of F_{j_1,j_0}, the translation of m and the optimal value for s can all be calculated in order to decide whether to retain the polynomial.

3.4.2 Selecting better polynomial pairs from a set

Once we have used one of the procedures described above to produce a set of polynomial pairs with attractive properties we need a way to choose the most appropriate polynomial pair from that set. Generally the procedures produce far too many polynomials to conduct sieving experiments on and so some other ranking system must be provided.

In order to do this Murphy made use of $\alpha(F)$; we require a fairly good estimate of $\alpha(F)$ at this stage and since the small primes can have such a large effect on the value of $\alpha(F)$ we must take care with the those small primes which are not well behaved. Murphy computes $\text{cont}_p(F)$ for these primes by counting appearances of $p^k, k \geq 1$ in a sample of F values. We then take the mean. For the larger primes $\text{cont}_p(F)$ is just estimated as usual. Murphy considers the small primes to be those below 100 and in addition computes estimates for $100 < p < 2000$.

The key idea is that the set of polynomials is ranked based on a raw estimate of the differences in yield. The ranking is considered to be independent of variations in B, the smoothness bound, but cannot be used to compare polynomials of differing degrees (or pairs of polynomials whose degree sum differs).

Again we must separate the discussion into non-skewed and skewed, we follow Murphy's descriptions.

Non-Skewed

In this case the ranking is determined in the most part by the non-linear polynomial F_1 as the plausible m values are of a similar size. Let us consider a way of ranking a single homogeneous polynomial $F_i(X,Y) = Y^d f_i(X/Y)$. In polar coordinates we have

$$F_i(X,Y) = r^d F_i(\cos\theta, \sin\theta)$$

thus if we fix θ any two polynomials of the same degree d grow as the dth power of r along θ. Hence the values $F_i(\cos\theta, \sin\theta)$ are the most relevant for ranking the yield.

We recall that the function $\rho(1/\beta)$, $\beta = \log B/\log x$ can be used to gain a rough approximation of the number of B-smooth integers up to x. For $j = 1, \ldots, k$, $\theta_j = \pi/k(j - 1/2)$ we calculate

$$\beta_{F_i}(\theta_j) = \frac{\log B_i}{\log|F_i(\cos\theta_j, \sin\theta_j)| + \alpha(F_i)}$$

where the θ_j are the mean values of k equally sized sub-intervals of $[0, \pi]$ and then the polynomial F_i is given the rating

$$\mathbb{E}(F_i) = \sum_{j=1}^{k} \rho(1/\beta_{F_i}(\theta_j)).$$

The polynomials in the set are ranked in descending order. The value of k is said not to be crucial, Murphy uses $k = 1000$.

Of course we need to take both polynomials into account so the rating that is actually used is

$$\mathbb{E}(F_1, F_2) = \sum_{j=1}^{k} \rho(1/\beta_{F_1}(\theta_j))\rho(1/\beta_{F_2}(\theta_j)).$$

Skewed

When working with a skewed non-linear polynomial we use a skewed sieve region with length to width ratio given by s. In addition to this we must always work

with both polynomials as the m-values can differ significantly. A generalisation of $\mathbb{E}(F_1, F_2)$ works with a skewed sieve region, in fact we work with an ellipse with major and minor axes in ratio s defined by $X = \sqrt{s}\cos\theta$, $Y = 1/\sqrt{s}\sin\theta$, $\theta \in [0, \pi]$. Dividing the interval into k equally sized sub-intervals and working with θ_j, the mean of each interval we define

$$\beta_{F_i}(\theta_j) = \frac{\log B_i}{\log|F_i(\sqrt{s}\cos\theta_j, \frac{1}{\sqrt{s}}\sin\theta_j)| + \alpha(F_i)}$$

and take

$$\mathbb{E}(F_1, F_2) = \sum_{j=1}^{k} \rho(1/\beta_{F_1}(\theta_j))\rho(1/\beta_{F_2}(\theta_j)).$$

We again rank the polynomial pairs in descending order of $\mathbb{E}(F_1, F_2)$ ratings.

In both the skewed and non-skewed cases we would then choose subsets of pairs F_1, F_2 with high ranking; to make the ultimate selection sieve tests may now be used.

3.5 Summary

We have summarised the standard methods of estimating the quantity of smooth integers below a bound and we have considered the key criteria that affect yield in the case of the number field sieve. Further to this we have seen how the probability of smoothness of a number encountered in the sieving step can be tied to the probability of smoothness of a random number of the same size. We have seen how these ideas have been used to:

1. Estimate the total yield in order to compare two variants of the number field sieve.
2. Provide the basis for methods to select improved general number field sieve polynomials.

These are key ideas that we will return to at various points in the remainder of this thesis.

Chapter 4

Estimating yield

As briefly introduced in the preceding chapter, Cavallar [14, 20] suggested a method of estimating yield over the whole sieve region — including the large prime relations. However, Cavallar's method produces a significant underestimate on the non-linear side: we investigate the reasons for this with a view to improving the method.

The driving force behind this work is the desire for a more robust method of evaluating possible variants of the number field sieve; in particular we will require a method to estimate the quantity of relations produced in various SNFS factorisations. It is hoped that a more reliable method of estimating the yield in this case, alongside sieving tests and theoretical predictions, will support the assessment of a proposed SNFS variant in chapter 6 of this thesis.

4.1 Cavallar's method

Cavallar used the approximations G_i, H_i introduced in the previous chapter to estimate the quantity of full and partial (1, 2 or 3 large primes on each side) relations produced by various factorisations with a view to assessing the three large primes variant of the number field sieve.

Let $\Psi(x, y)$ and $\Psi_i(x, y, z)$ be as previously defined. Recall the assumption that

the polynomial values $F(a,b)$ are B-smooth with the same probability as randomly selected integers r with logarithmic norm $\log F(a,b) + \alpha(F)$ where

$$\alpha(F) = \sum_{p \leq B} (\mathrm{cont}_p(r) - \mathrm{cont}_p(F)) \log p.$$

We join Cavallar in making the following assumptions some of which we will discuss further presently:

1. We use the classical sieve. There is no clear way to produce estimates of this type for the lattice sieve, however as we wish to use this as a mechanism to compare variants of the number field sieve and not to estimate the outcome of particular parametrisations this assumption is not too confining.

2. The sieving region is usually $R_c = [-a, a] \times [1, b]$, the set of auxiliary numbers to be tested for smoothness is therefore $R = R_c \cap \mathbb{Z} \times \mathbb{N}$.

3. $\Psi(x,y)/x$ is the proportion of y-smooth numbers amongst the numbers from 1 to x. The average size of these numbers is $(1+x)/2$ which we note is approximately $\bar{x} = x/2$.

4. The mean size of $F(X,Y)$ over the continuous R_c is

$$\bar{F} = \frac{\iint_{R_c} |F(X,Y)| dX dY}{\iint_{R_c} dX dY}.$$

   This implies that we may treat the values $|F(a,b)|$ like random values of mean size $\bar{x}' = \bar{F} e^{\alpha(F)}$ that is, we would use $\Psi(x', B)/x'$ with $x' = 2\bar{x}'$ to approximate the portion of B-smooth polynomial values among the pairs (a,b) from R.

5. In the sieving region we have approximately $Z = 6/\pi^2 \iint_{R_c} dX dY$ pairs such that $\gcd(a,b) = 1$ [54, section 4.5.2]. Hence we expect $Z\frac{\Psi(x',B)}{x'}$ B-smooth norms amongst them and $Z\frac{\Psi_j(x',L,B)}{x'}$ (j,L,B)-smooth norms.

6. We assume the polynomials are independent — that is, we assume the probability of smoothness of $F_1(a,b)$ is independent of the smoothness of $F_2(a,b)$, in fact this is not the case as there are minor effects for primes dividing the resultant of the two polynomials. We will follow Cavallar in

ignoring these effects and hence assume that we may use

$$Z\frac{\Psi_{j_1}(x_1', L_1, B_1)}{x_1'}\frac{\Psi_{j_2}(x_2', L_2, B_2)}{x_2'}$$

as an approximation of the number of j_1, j_2-partial relations.

7. The single large prime bound L is less than or equal to B^2 (this assumption allows us to recognise single large prime relations essentially for free in the sieve). The double large prime bound is greater than B^2 but less than or equal to B^3 (integers between these bounds will have a maximum of two primes larger than B in the prime decomposition).

We recall the introduction in the previous chapter of the functions G_i, $i \in \mathbb{Z}$, $i \geq 0$. For $0 < \alpha < 1$:

$$G_0(\alpha) := \lim_{x \longrightarrow \infty} \frac{\Psi(x, x^\alpha)}{x} = \rho\left(\frac{1}{\alpha}\right).$$

More generally we had that if $0 < \alpha < \beta < 1/i$ then

$$\begin{aligned} G_i(\alpha, \beta) &:= \lim_{x \longrightarrow \infty} \frac{\Psi_i(x, x^\beta, x^\alpha)}{x} \\ &= \frac{1}{i!}\int_\alpha^\beta \cdots \int_\alpha^\beta \rho\left(\frac{1 - (\lambda_1 + \cdots + \lambda_i)}{\alpha}\right)\frac{d\lambda_1}{\lambda_1}\cdots\frac{d\lambda_i}{\lambda_i} \end{aligned}$$

Based on the above assumptions and formula Cavallar estimated the quantity of full and partial relations in the following manner:

For $i = 1, 2$, with number fields K_i, defined by polynomials F_i, with factor base bounds B_i and large prime bounds L_i we let $x_i' = 2\bar{F}_i e^{\alpha(F_i)}$, $\alpha_i = \log_{x_i'} B_i$, $\beta_i = \log_{x_i'} L_i$ and calculate

$$ZG_{j_1}(\alpha_1, \beta_1)G_{j_2}(\alpha_2, \beta_2)$$

to estimate the quantity of j_1, j_2-partial relations.

In order to accomplish this the Dickman function ρ must be calculated to high precision. This is accomplished as described in chapter 3.

We might ask what we are able to discern when α and β are not in the bounds

described above. If B is our smoothness bound and we consider the interval $[1, x]$ then we have the following:

If $\alpha \geq 1$ then

$$\alpha = \frac{\log B}{\log x} \geq 1 \;\Rightarrow\; \log B \geq \log x \;\Rightarrow\; B \geq x$$

and in this situation it is clear that all integers in the interval $[1, x]$ are B-smooth.

If $\alpha < 1$, $\beta \geq 1$, $\beta = (\log L / \log x)$ where L is the large prime bound then we can deduce by the same argument that all integers in $[1, x]$ are L-smooth and may use the function G_0 to estimate the quantity of these that are full relations.

If $\alpha \geq 1/2$ then

$$\alpha = \frac{\log B}{\log x} \geq \frac{1}{2} \;\Rightarrow\; \log B \geq \frac{1}{2}\log x \;\Rightarrow\; B \geq x^{1/2} \;\Rightarrow\; B^2 \geq x$$

since all single large prime partial relations must have the large prime below B^2 we can deduce that all integers in the interval are single large prime relations or full relations. We may use G_0 to estimate the quantity of these that are full relations. By a similar argument we may deduce that all integers in the interval must be at least two-partials if $\beta \geq 1/2$. These small results allow us to work with intervals or regions in which $\alpha, \beta > 1/2$ which will be necessary if we encounter quite small regions that contain, for one reason or another, small polynomial values.

Cavallar found that the estimates were between 44% and 74% of the actual quantity produced. More interestingly the data for the number $2^{773} + 1$ showed that the approximation was radically better on the linear side than on the non-linear side. In this extended example the linear and non-linear sides were estimated separately so we are better able to see how the underestimate is produced. On the linear side the estimates produced were extremely good: those made using x' were within 1% using the G_i defined above (based on the lower estimate of $\Psi(x, y)$) and within 7% using the equations H_i based on the less simplistic estimation of $\Psi(x, y)$ mentioned in the preceding chapter. On the other hand the approximations on the non-linear side were up to 66% off.

Cavallar notes that the linear polynomial is near constant over the region while the degree 6 polynomial increases from $2 \cdot 10^{36}$ to $5 \cdot 10^{44}$. It would appear that the single average taken on the non-linear side cannot adequately represent the wide range of values. Cavallar suggested that it is likely that you could improve these estimates by splitting the sieving region into smaller parts and calculating the estimates over these.

We intend to test this hypothesis. The most immediate question regards how we should split the sieving region in order to achieve an improved estimate. It is not immediately obvious whether the estimate produced will be dependent on the method as we have not established the exact nature of this underestimate.

4.2 Towards an explanation of the underestimate

If we accept the assumption that $\rho(1/\alpha)$ is a good approximation for $\Psi(x, x^\alpha)$ as analysed in [51] and hence that the G_i, $i > 0$ are good approximations for $\Psi_i(x, x^\beta, x^\alpha)/x$ as analysed in [20, section 2.6] and, in addition, believe the method of calculating high precision approximations for $\rho(1/\alpha)$ is sound then we must look elsewhere in Cavallar's technique in order to isolate the cause of the underestimate.

It is certainly possible that the source of the underestimate is in fact the underlying approximation for $\Psi(x, x^\alpha)$. However, it seems reasonable to accept these assumptions given the analysis and that, with $\alpha(F)$ defined as above, they have led to adequate estimations of yield in the case of intervals and the quadratic sieve. We will work on the basis that these assumptions are acceptable and leave further analysis of whether they are sound to others.

The assumption that the polynomials are independent is unlikely to be implicated as the effect is seen when considering estimations involving only one polynomial.

Boender [7] and Murphy [75, 76] both used the interval method of approximating the quantity of full relations in the situation where $b = 1$. There is no obvious counterpart to this method for regions $[-a, a] \times [1, b]$, $b > 1$. Cavallar's method is

actually estimating the proportion of integers between 1 and x' which are smooth where $x' = 2\bar{F}e^{\alpha(F)}$ and $\bar{F}$ is the assumed to be an appropriate approximation of the mean of the integral values taken by $|F(X,Y)|$ on R_c. Clearly we require not only that $\bar{F}$ is fairly representative of the integral values actually taken over the sieve region but that other descriptive statistics suggest that the smoothness properties of integers in the set R can be adequately assessed by consideration of the smoothness properties of integers in the interval $[1, x']$. There are various concerns:

1. $\bar{F}$ is not equal, in general, to the mean or median of the integral points on $|F|$: as degree of F increases or region size decreases it is possible that this may be significant.

2. If the values of $|F|$ have a large range then the ability of any average to represent the data is more likely to be impaired (though the range can be sensitive to extreme values so this may not be the case).

3. The distribution of integral values taken by $|F|$ may be particularly skewed. In this case the mean can be significantly different from the median. Since this is not the case in the interval $[1, x']$ this could be a cause for concern.

The method could be further compromised by the assumption that we may simply multiply by $Z = 6/\pi^2 \iint_{R_c} dXdY$ in order to find the estimate only for pairs (a, b) such that $\gcd(a, b) = 1$.

We aim to improve the estimates. A method suggested by Cavallar is to split the sieve region into smaller subregions, make the estimate in each case and then sum over these. Unfortunately the success of such a method could rely substantially on the method of splitting chosen.

In the case of intervals, Boender and Murphy split the interval in the following manner:

The interval is split into segments within which the function f has no roots or turning points hence, $|f|$ is increasing or decreasing on each segment. Let f_i be the continuous function on the i^{th} segment. As we work identically on each segment, let us consider only the k^{th} segment. Define $S_1 = \min(f_k)$, $S_2 = \max(f_k)$, cut

the segment into K subintervals by taking

$$h = \frac{\log S_2 - \log S_1}{K}$$

and set $y_j = S_1 e^{jh}$, $j = 0, \ldots, K-1$. We then define the subintervals to be $[x_j, x_{j+1}]$ for which $f_k(x_j) = y_j$, $f(x_{j+1}) = y_{j+1}$. Hence we split the function into subintervals on which the values taken by the function are of similar size.

It is not clear how such a method could be generalised to regions in an efficient manner and this would appear to be an open question.

Before suggesting a more appropriate method of splitting the region we must clarify which assumption is responsible for the underestimation. We will assess the linear and non-linear sides separately to aid us in this.

4.3 The linear side

Prior to considering the more complex case of a region $[-a, a] \times [1, b]$ we will consider Cavallar's method with $b = 1$; in this case we remove one assumption as we do not need to multiply by Z (since $\gcd(a, 1) = 1 \quad \forall a$). This will allow us a direct comparison with the approach taken by Murphy and Boender and it may also allow us to quantify what is most important in producing reliable estimates.

4.3.1 Intervals

The first experiment is based on that carried out by Boender [7]. We will use Cavallar's method to estimate the yield when sieving the numbers in an interval. This would never arise from use of the number field sieve but it should allow us to deduce the impact of using an interval $[1, 2\bar{x}]$ to represent an interval $[x, x+\Delta]$ with continuous mean $\bar{x}$.

Boender tested the interval estimation technique on the intervals $[x, x + \Delta]$ with smoothness bound y as defined below and we use Cavallar's method to produce estimates for the same parameters. Boender's interval estimation function utilises

a more sophisticated approximation than the G_i functions — the H_i functions mentioned in the previous chapter. So we will also consider Cavallar's method using the function H_0.

				Boender's Estimate [7]		Cavallar's G_0 Method	
x	y	Δ	Actual	Est.	Quo.	Est.	Quo.
10^{27}	5×10^4	$10^8 + 2 \times 10^5$	3521	3606	1.024	3473	0.99
10^{35}	3×10^5	$10^8 + 2 \times 10^5$	529	527	0.996	505	0.96
10^{40}	8×10^5	$10^8 + 2 \times 10^5$	149	159	1.067	152	1.02
10^{45}	6.5×10^5	$10^{11} + 2 \times 10^6$	6818	6771	0.993	6483	0.951
10^{50}	8.5×10^5	$10^{11} + 2 \times 10^6$	666	646	0.970	619	0.929
10^{50}	10^6	$10^{11} + 2 \times 10^6$	928	912	0.983	873	0.941

Considering that the estimate is produced using only the mean of the values in the interval the results are surprisingly good. Taking the H estimates instead of the G estimates we find that we tend to overestimate but produce comparable results:

				Boender's Estimate [7]		Cavallar's H_0 Method	
x	y	Δ	Actual	Est.	Quo.	Est.	Quo.
10^{27}	5×10^4	$10^8 + 2 \times 10^5$	3521	3606	1.024	3876	1.100
10^{35}	3×10^5	$10^8 + 2 \times 10^5$	529	527	0.996	558	1.055
10^{40}	8×10^5	$10^8 + 2 \times 10^5$	149	159	1.067	167	1.12
10^{45}	6.5×10^5	$10^{11} + 2 \times 10^6$	6818	6771	0.993	7169	1.051
10^{50}	8.5×10^5	$10^{11} + 2 \times 10^6$	666	646	0.970	685	1.029
10^{50}	10^6	$10^{11} + 2 \times 10^6$	928	912	0.983	965	1.04

In the interval linear case the mean is equal to the median of the data, the range is as small as can be; there are no extreme values and no skew.

However, we also need to consider linear polynomials, such as $a_1X - a_0$, of the form we might reasonably expect to encounter in the number field sieve. We will first consider the case where $b = 1$ and then proceed to extend our results to the region R.

					G		H	
$f(X)$	$\alpha(f)$	B	a	Actual	Est.	Quo.	Est.	Quo.
$X - 2^{43}$	0.0	3572	4800	99	91	0.919	103	1.04
$X - 2^{90}$	0.0	$8.1 \cdot 10^5$	$3.5 \cdot 10^5$	681	668	0.981	724	1.063
$3^{55}X - 1$	0.57	$4.4 \cdot 10^6$	$1.68 \cdot 10^6$	1545	1469	0.951	1581	1.026

As can be seen in the table above the results are extremely good in these situations. It appears that at least on the linear side this method of estimating yield may require little improvement. We will note various descriptive statistics regarding these test cases in order to throw more light on the results.

In each linear case considered above the continuous mean, the discrete mean and the median are equal. This is due both to the linear nature of the polynomials involved and to the size of the single large coefficient in the linear polynomial. In NFS this linear coefficient will be of the size $n^{\frac{1}{d+1}}$. As long as this dominates the polynomial (and the range) the skew will be 0. We put forward the hypothesis that it is the equality between the mean and the median (or, equivalently, the lack of skew) that allows the method of estimation to produce such a good estimate in the linear case.

4.3.2 Regions

We continue this section by extending these observations to the case of regions of the type R before going on to investigate the non-linear side in the next section where we will produce further evidence that the error in our estimations is correlated with the skew.

In the following all $\alpha(F)$ values are 0.570, X ranges in the interval $[-a, a]$ and Y in $[b_1, b_2]$; B is our smoothness bound.

$F(X,Y)$	B	a	$[b_1, b_2]$	Actual	Est.	Quo.
$X - 2^{43}Y$	3572	4800	$[1, 2000]$	12847	10962	0.853
$X - 2^{90}Y$	$8.1 \cdot 10^5$	7000	$[1, 2000]$	3943	3589	0.910
$X - 2^{129}Y$	$2 \cdot 10^7$	28875000	$[10^6 + 1, 10^6 + 100]$	36214	36220	1.000

Since the range is so much larger in these cases we might expect that the continuous mean would be less able to provide for a useful estimate; however the estimates are still reasonably good. This provides further evidence for our hypothesis — that it is not only the range of values that is of importance but their distribution.

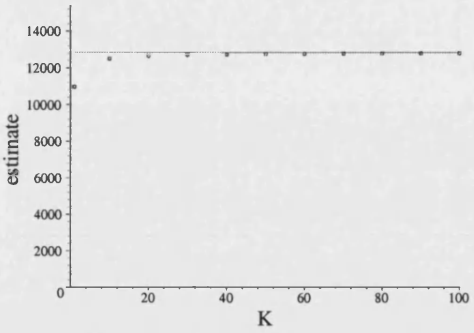
Certainly, on the linear side Cavallar's method appears to produce reasonable estimates and hence there is little evidence to suggest that we need to split the sieve region in order to improve the estimates. On the other hand if we do need to split the sieve region in order to adequately estimate the non-linear side we will eventually need to use the same splitting method on the linear side (since in reality we do not treat the polynomials separately). In addition we would be interested to see if there is likely to be any beneficial effects on results that are already good. In fact, splitting the sieve region into equal sized subregions does have some effect as we see in figure 4.1 particularly if the polynomial values involved are not particularly large. However it is by no means worrying, the results are in fact quite favourable in the examples given and other linear polynomials behave in a similar manner. One question that immediately arises at this point and which will remain with us throughout the rest of this chapter is the issue of how far to split the region. If the method used to split the region does not produce a bounded result or does not naturally terminate in some way then how can we be certain that the method can be put to good use? In the linear cases the method appears to reach a conclusion.

Splitting the a and b intervals into K subintervals (and hence splitting the region into $K \times K$ equally sized rectangular regions) we find that we achieve very commendable results with a reasonably small values K and that the estimates produced appear to be bounded by the actual yield.

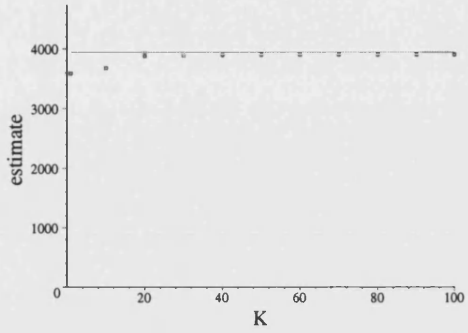
In the case of the polynomial $X - 2^{129}Y$ all the splits produced the quantity of relations 36220 as an estimate for the actual quantity 36214; hence this tiny overestimate appeared to be stable.

However, it seems possible that splitting the sieve region equally will not function as well in the non-linear case where we cannot assume that the skew is low enough for the means in equally split regions to be useful. To illustrate the effect of skew on the estimates produced we will return to the interval case.

Figure 4.1: Two examples of estimates, using equal splits of the sieve region into $K \times K$ subregions, approaching the actual quantity of sieved relations as K increases.



$X - 2^{43}Y$, $B = 3572$, $|a| \leq 4800$, $1 \leq b \leq 2000$



$X - 2^{90}Y$, $B = 8.1 \cdot 10^5$, $|a| \leq 7000$, $1 \leq b \leq 2000$

4.4 The non-linear side (intervals)

We will consider a unique situation that will help us to quantify the precise problem in Cavallar's method. Let $f_5(X) = X^5 - 3$, $f_4(X) = 10^8X^4 - 3$, $f_3(X) = 10^{16}X^3 - 3$ and $f_2(X) = 10^{24}X^2 - 3$ and sieve over the interval $[0, 10^8]$. Over this interval the four polynomials take values that lie in the range $[-3, 10^{40} - 3]$ but otherwise are very different. We will use the same prime bound of 2×10^6.

We are interested in assessing the ability of the average to act as a representative of the data in situations that are not akin to those found in the linear case and the effect that this has on the estimate.

We work as Cavallar does to approximate the values of $\alpha(f)$ (except we work with the formula for $\alpha(f)$, not $\alpha(F)$). That is, we use random sampling and calculation to estimate $\text{cont}_p(f)$ for $p|\,\text{Disc}(f)$. For all the other primes we factor $f \bmod p$, count the roots and divide by $p - 1$ to get $\text{cont}_p(f)$. We allow sampling across $[0, 10^8]$ and use a factor base bound of 10000 (smaller factors almost entirely control the value of $\alpha(f)$). The random sampling needed to find $\text{cont}_p(f)$ means that we can experience a small range of reported α values, (we average 5 reports).

$f(X)$	$\alpha(f)$	mean	Act.	G Est.	G Quo.	H Est.	H Quo.
$X^5 - 3$	0.661	$1/6 \cdot 10^{40} - 3$	5059	747	0.148	814	0.161
$10^8X^4 - 3$	1.677	$1/5 \cdot 10^{40} - 3$	1713	580	0.339	633	0.370
$10^{16}X^3 - 3$	0.635	$1/4 \cdot 10^{40} - 3$	1322	689	0.521	751	0.568
$10^{24}X^2 - 3$	1.638	$1/3 \cdot 10^{40} - 3$	666	523	0.785	571	0.857

As we can see, it is fairly easy to produce a situation with $b = 1$, in which Cavallar's method is highly misleading. Let us consider why the method is so poor in this case. The range in each case is equal (although this is due to extreme values in three of the four cases). The first difference is the mean, the worst estimates occurring when the mean is smallest. However the means are all numbers of approximately the same size. If this is in fact the cause then it does not seem likely that we will be able to improve the estimates. There is one other key difference between the four examples however and that is the discrepancies between the continuous mean used to calculate the estimates, the discrete mean of the integral polynomial values over the interval and the median polynomial value. If we consider the first example, the continuous mean used is $1/6 \cdot 10^{40} - 3$ while the discrete mean is in fact

$$\frac{1}{6} \cdot 10^{40} + \frac{1}{2} \cdot 10^{32} + \frac{5}{12} \cdot 10^{24} - \frac{1}{12} \cdot 10^{8} - 3/10^{8},$$

and the median is $\frac{1}{32} \cdot 10^{40} - 3$. This latter is far more significant — the median is actually an order of magnitude smaller than both the discrete and continuous means, suggesting that this is a set of polynomial values with a skewed distribution. The effect becomes less and less drastic as we move down the table. This supports the theory that the main fault in the method is the inability of the mean to adequately represent the data due to either a larger range (ignoring extreme values) or a dataset with a skewed distribution.

Clearly we need to split the sieve interval/region in the non-linear case to produce realistic estimates but it seems increasingly likely that splitting into equal intervals/regions will not be the most appropriate method to deal with the problem at hand. We will continue to accumulate evidence that no mean can effectively represent the data due to inherent skew by trying to split the sieve interval equally in the next section. We aim to provide additional evidence that we need to split the interval in such a way that the skew is minimised.

4.4.1 Splitting the sieve interval

There are some issues with splitting up the sieve interval in order to improve the estimate of which we should be aware:

- As the estimate for each subinterval would be subject to error, we could be increasing the size of the error by splitting up the interval.
- If the process does not tend toward some bound we would have no idea how many intervals we need to work with. We require a method that reaches a natural conclusion in some manner. In addition, if we must choose, it is better to produce what we reliably know to be an underestimate than to produce an overestimate that we do not know to be bounded.
- The estimate may be sensitive to how the interval is split up.
- It appears that the linear side does not *require* splitting (but can benefit from it). Depending on how we split up the intervals we may have to split the intervals in this way for all of the polynomials involved. This would also impact on the usefulness of the method for assessing variants such as the multi-polynomial number field sieve.

We will compare three methods of splitting the sieve interval as it is not immediately clear that the results achieved will not be heavily reliant on the method used. We will also use only methods that can be reliably generalised to the region case.

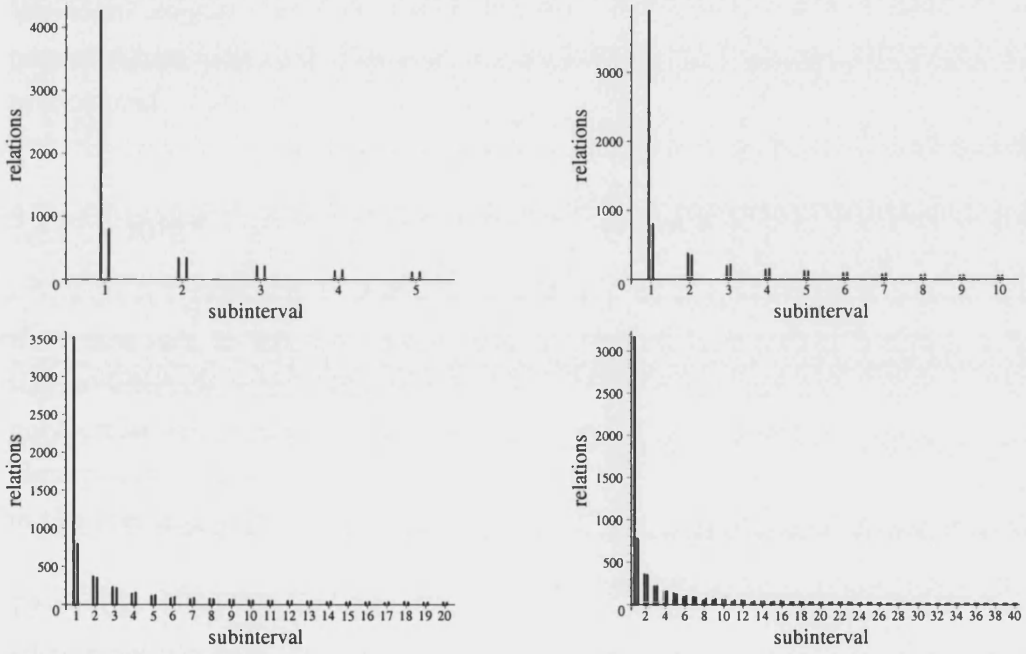
Let us consider different ways in which we could split the interval in order to produce a less misleading estimate. The first and obvious method to try is simply to split the interval into K equal parts. The key advantage of this method is that it will translate simplistically to the region case. The disadvantage is clear — we are not dealing with the root cause of the problem. If the size of the range is the issue then those intervals further from $a = 0$, where the polynomial values are larger, will be worse affected. If it is the size of the range with respect to the the size of the polynomial values we would expect the method to work equally well across all of the intervals. In the case that it is the skew that is causing the problem we would expect the worst effects to be in those intervals closest to $a = 0$ since the smaller size of the polynomial values will amplify the effect.

However we will first attempt to improve matters by equal splitting, and after noting any problems with this technique we will consider ways we might adapt splitting to the particular polynomial or interval.

$f_5(X) = X^5 - 3$:

As we can see from figure 4.2 the estimates for $f_5(X) = X^5 - 3$ are excellent in all subintervals except the first.

Figure 4.2: Actual quantity of relations followed by estimated quantity in each subinterval for $K = 5, 10, 20, 40$ highlighting problems in the first subinterval.



Let us consider the values that f_5 takes over the first subinterval in each case:

K:	5	10	20	40
range:	$32 \cdot 10^{35}$	10^{35}	$3125 \cdot 10^{30}$	$25^5 10^{25}$
cts mean:	$\frac{32}{6}10^{35} - 3$	$\frac{1}{6}10^{35} - 3$	$\frac{3125}{6}10^{30} - 3$	$\frac{25^5}{6}10^{25} - 3$
median:	$10^{35} - 3$	$\frac{1}{32}10^{35} - 3$	$\frac{3125}{32}10^{30} - 3$	$\frac{25^5}{32}10^{25} - 3$

While the range shrinks substantially as K grows the ratio between the continuous mean used in our calculations and the median of the integer values taken on the

polynomial remains equal to 32/6 in the first interval. Thus we are in danger of overestimating the size of the polynomial values, this would in turn cause an underestimate of the probability of the values taken on the interval being B-smooth.

In the subsequent intervals this effect is not nearly so severe. If we consider the second interval in the case $K = 5$ then we have a range of $992 \cdot 10^{35}$, which is in fact larger than in the first interval. We use the continuous mean $336 \cdot 10^{35} - 3$ and the median is $243 \cdot 10^{35} - 3$ giving us a ratio of 1.38. The remaining intervals have larger range again but the average used to calculate the results approaches the median suggesting that the polynomial values are less skewed.

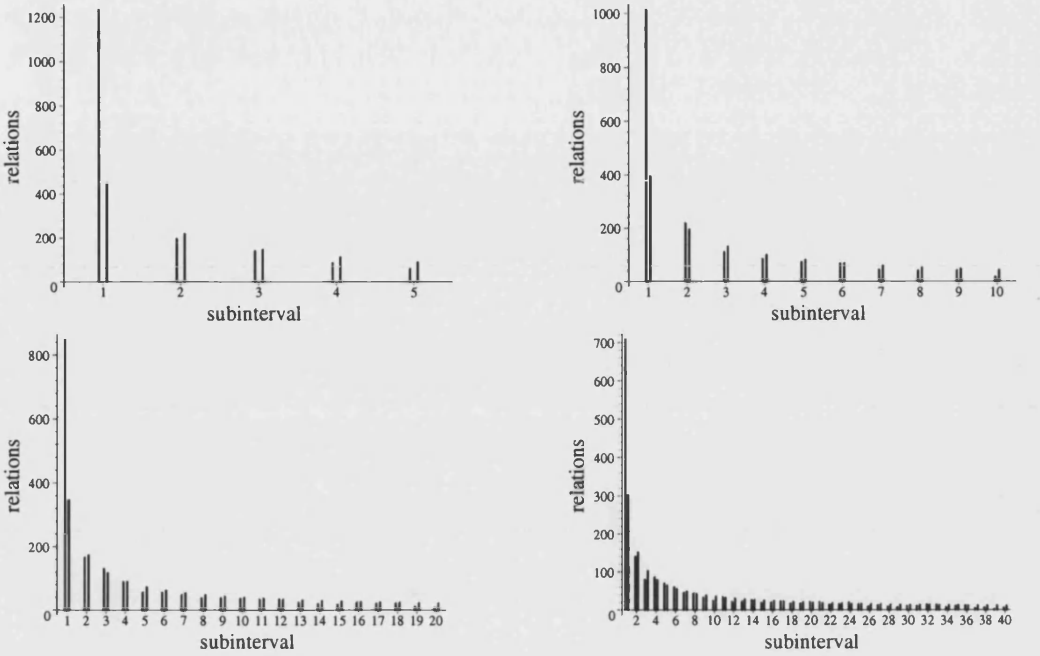
We might suggest that this effect is due to the extremely small size of the polynomial values near to 0. However, we see precisely the same effect (though less pronounced) when working with the polynomial $f_4(X) = 10^8X^4 - 3$.

$f_4(X) = 10^8X^5 - 3$:

If we now turn to figure 4.3 we see that we have a similar state of affairs again: the subintervals worst affected are those closer to 0, however in this case the polynomial values taken in this interval do not have extremely small size. The discrepancy between the mean used and the median is less severe (a ratio of 3.2 in the first interval).

This provides further evidence that the inability of the continuous mean used to adequately represent the polynomial values taken in the interval or region is the cause of the significant underestimate. In addition it appears to suggest that it is not the range of values taken that is the chief cause as the range is actually smallest on the worst affected intervals. Worryingly the significant underestimate is most likely to occur close to $a = 0$ and this is also one area of the sieve interval that we would expect to produce a large yield. This could cause problems with estimation, particularly in the special cases which have polynomials with extremely small coefficients — and hence are particularly productive close to $a = 0$.

Figure 4.3: Actual quantity of relations followed by estimated quantity in each subinterval for $K = 5, 10, 20, 40$ highlighting problems in the first subinterval.



4.4.2 Improved methods for splitting the interval

When we split the linear polynomial $|F|$ over the sieve region the estimate was reasonably good. This is regardless of the size of numbers involved or the range of number involved. If we split the non-linear side into equal subparts we find that we produce a significant underestimate in some of the intervals. This effect seems to be intimately connected with the ability of the continuous mean to represent the data and in particular, when the median and the mean differ significantly we produce extremely poor results.

Ideally we would like a method for splitting the interval or region that naturally addresses this difficulty. It is clear that we cannot place a large amount of confidence in the results produced so far.

The most natural approach is a method that adapts the split into subregions based on the properties of the polynomial, perhaps taking smaller subregions where the polynomial is changing most rapidly for instance. This approach is similar in nature to numerical integration. However, splitting the sieve region in this way is quite complex and it would not be possible to choose to split the region

in to a precise quantity of subregions — making it harder to draw comparisons between results if we could not also produce error bounds on the estimates. It is possible that we would be able to achieve this eventually but without a substantial collection of full sets of sieved data it becomes difficult to assess such a method. Eventually an approach which echos known adaptive quadrature methods may be the most appropriate and we will return to this question when considering further work. We aim now to test a more general idea: that we may improve the estimates by reducing the effect of the skew. We will use the more unsophisticated approach of taking smaller intervals where the difference between the continuous mean and the median is likely to be most significant. Since in general we will not be able to calculate the actual median over a region and hence cannot be sure of the actual skew over the sieve area we must work at a high level of generality. That is, we note that this effect is likely to be strongest in areas near to the origin. By creating smaller intervals in these areas we may be able to manage the effect and hence produce a more stable estimate.

We consider two possible methods in the interval case. In both we take the smallest intervals close to 0, where the change in size of polynomial values is most significant. As we are not splitting the interval into equally sized subintervals we are more likely to have subintervals that consist only of integers below the various smoothness bounds. In this case we will may not be able to use all the estimates G_i but rather can calculate more directly the probability of the polynomial values being B-smooth.

Method 1:

1. If the interval crosses $a = 0$ and we require an even number of intervals then split the interval into two at $a = 0$ and then work on the two subintervals. If K is odd, allow a small interval about 0 which we will not split any further and then continue to work on what is left separately.

2. We may now assume that the a values are strictly increasing or strictly decreasing in any interval we wish to split. Assume the former; we are working in an interval defined by $[a_{\min}, a_{\max}]$, $a_{\min}, a_{\max} \geq 0$.

3. If $a_{\min} \neq 0$ then let $start = \log(a_{\min})$ otherwise set $start = 0$. Let $stop = \log(a_{\max})$, set $w = (stop - start)/K$ and define our intervals by $a_i = a_{\min} +$

$\exp(iw)$, $[a_i, a_{i+1}]$, $0 \leq i \leq K-1$.

Clearly this simplistic method can be utilised in the case of regions; we only need to split into rectangular regions by working in the same manner along the b interval and the a interval.

This method is reminiscent of that used by Boender and Murphy however in their case they split the Y-interval and not the X-interval, on the assumption that the size of the polynomial values was of importance. While this appears to be a correct assumption this method has no obvious counterpart in the case of regions — only with great difficulty can we isolate regions over which F has no zero valued points and no maxima or minima and hence we are unable to guarantee a split into a certain quantity of regions.

While our suggested method is rather simplistic in nature we hope to show that it produces improved results. We consider intervals first.

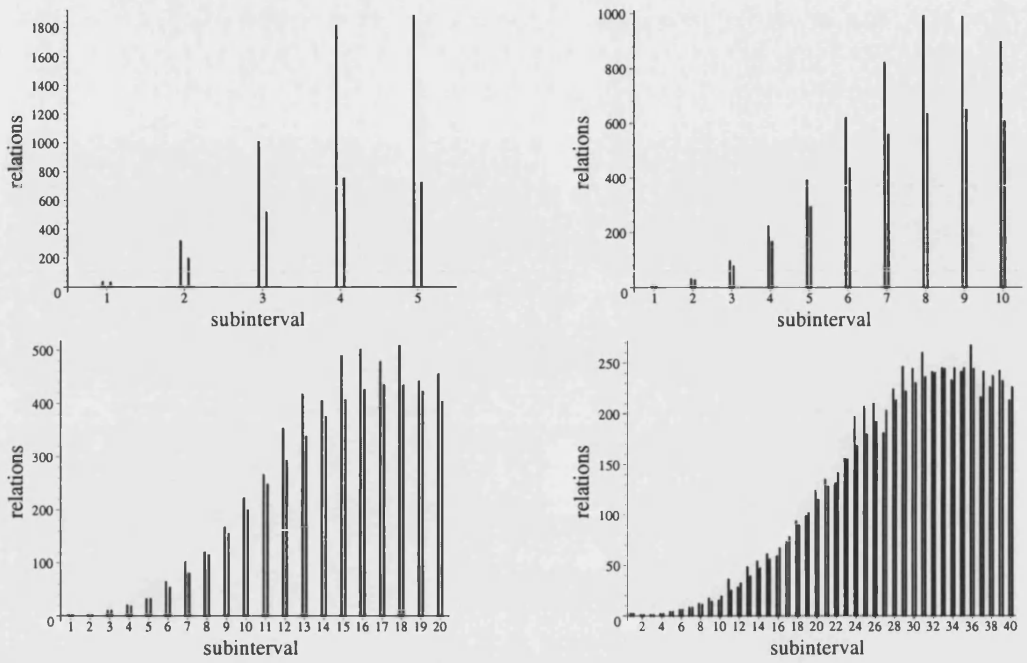
$f_5(X) = X^5 - 3$:

In figure 4.4 we return to an earlier example but calculate the estimates for each subinterval using method 1.

In this case the method requires a fairly large value of K in order to produce a good estimate in every subinterval. Computationally we would prefer to split the interval into as few regions as possible. The method does have some fairly attractive properties — as illustrated in figure 4.4 with a large enough value of K the estimates are significantly better than in the equal split; the method reaches a conclusion, that is, a maximal value of K (of a reasonable size) exists beyond which the method produces new intervals of zero length. However, for smaller values of K the method is far worse than splitting the interval equally. This is due to the way the split changes as we increase K.

We would ideally prefer a method that displays the positive qualities of the equal split — a fast convergence towards the result, improvements seen as K grows for small K and the positive qualities of method 1 — a maximal value of K and good estimates in all the subintervals. We suggest such a method.

Figure 4.4: Actual quantity of relations followed by estimated quantity in each subinterval for $K = 5, 10, 20, 40$ using method 1.



Method 2:

We split the interval $I = [a_{\min}, a_{\max})$ as follows:

1. If 0 lies in this interval and is not an endpoint then split the interval at 0. We will work instead with the subintervals $[a_{\min}, 0)$ and $[0, a_{\max})$ calling the method separately on each.

2. At this point all integers in I are either < 0 or ≥ 0; assume the latter. Split the interval at the points

$$a_i := (a_{\max} - a_{\min})/2^i, \quad i = 1, 2, \ldots K - 1$$

so we have the intervals $[a_{\min}, a_1), [a_1, a_2), \ldots [a_{K-1}, a_{\max}]$.

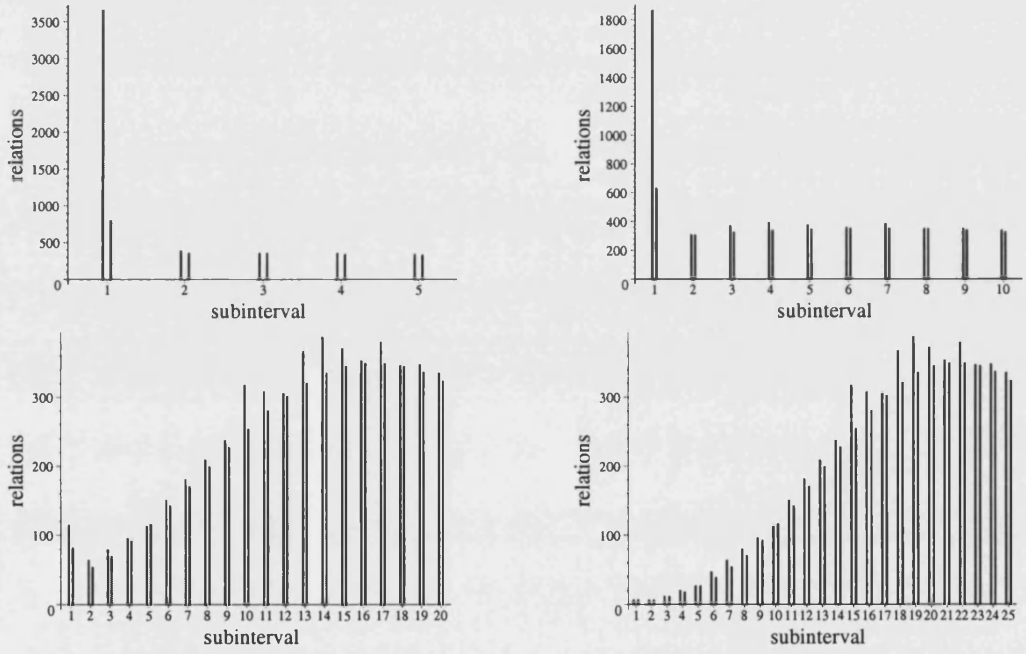
The purpose of this method is to combine the ideas behind splitting the interval equally — where we quickly improve the estimate except in the intervals close to 0 and method 1 above. In method 2 we focus on splitting into smaller intervals only those portions of the sieve interval that are most likely to cause a problem.

This method also has a maximal value for K, which is of a reasonable size, such that any larger value of K will produce intervals of zero length. In addition this method is likely to produce better results than method 1 for smaller values of K.

$f_5(X) = X^5 - 3$:

In figure 4.5 we return again to our example but calculate the estimates for each subinterval using method 2.

Figure 4.5: Actual quantity of relations followed by estimated quantity in each subinterval for $K = 5, 10, 20, 40$ using method 2.



This method is somewhat better on small values of K than the equal split method (note that for $K = 1, 2$ the methods are the same), due to the split the subinterval worst affected decreases in size far more quickly. We can also see that this method reaches an almost identical conclusion with larger K as method 1 but that it reaches this point for a smaller value of K (in fact the method has a smaller maximum K).

If we compare the actual estimates found in each case for $f_5(X) = X^5 - 3$ we find that both method 1 and method 2 produce reasonable estimates. The total yield is 5059, equal splitting with $K = 40$ gives the estimate 2742; method 1

with $K = 40$ produces the estimate 4942 and method 2, at the near maximal $K = 25$ gives 4712. The small size of the numbers involved means that we cannot conclude anything about the merits of method 1 over method 2 from this single result although the dismal performance of estimates produced by splitting the interval equally is noted.

Method 2 appears to display the positive qualities of the other methods as hoped, the smaller maximal value of K leads to less computational effort and we see better interim results for smaller K. We will see further support for this argument in the subsequent section where we work with sieve regions. Since there exist far better estimation methods in the case of intervals we will not pursue the matter further but move directly to the case of regions.

4.5 Splitting the sieve region

We will first compare the three methods for one factorisation in the hope that we can gain insight into whether splitting the region can produce an improved estimate.

We work in Maple and use the Maple code included in Lambert's thesis [56] as a starting point for the functions G_i. We note that a minor correction is needed to the output of G_2 which is in fact twice what it should be (this appears to be due to an incorrect change of variables in section 4.4 of Lambert's thesis).

Since we need to integrate the piecewise smooth functions $|F(X, Y)|$ which we typically work with we have also written Maple code to achieve this.

Comparison of the three methods

We compare the three methods which were investigated in the interval case. The factorisation used for the comparison was used by Cavallar [14, 20] and factored by Montgomery. The number is an Aurifeuillian factor of the form $3^h + 3^{\frac{h+1}{2}} + 1$ with $h = 331$. The parametrisation of the factorisation is found in 4.7 and 4.8 where the number is referred to as $3,993M$.

We have computed estimates of the quantity of j_1, j_2-relations for $0 \leq j_1, j_2 \leq 2$ using our three different methods. The graphs can be seen in figure 4.6. In each graph we have four curves, one for each of the methods and a horizontal line which marks the actual quantity of j_1, j_2-relations found by sieving. In each case, the curve that stops short is that of method 2, the other dotted black curve is that of method 1 and the lighter curve is the result obtained by splitting the region into equal parts.

In the case of equal splitting of the region we use $K \leq 38$, however, due to the nature of the method there is no means of deciding at which point it may be sensible to stop (the value of K at which we produce zero sized intervals is not only very large but implies that we take each interval to have a single point in it at the maximal value of K, this is not useful). In the case of the other methods, at the point at which one of the subregions has zero area we stop. Therefore for the first method we used up to $K = 38$ and for the second method up to $K = 20$.

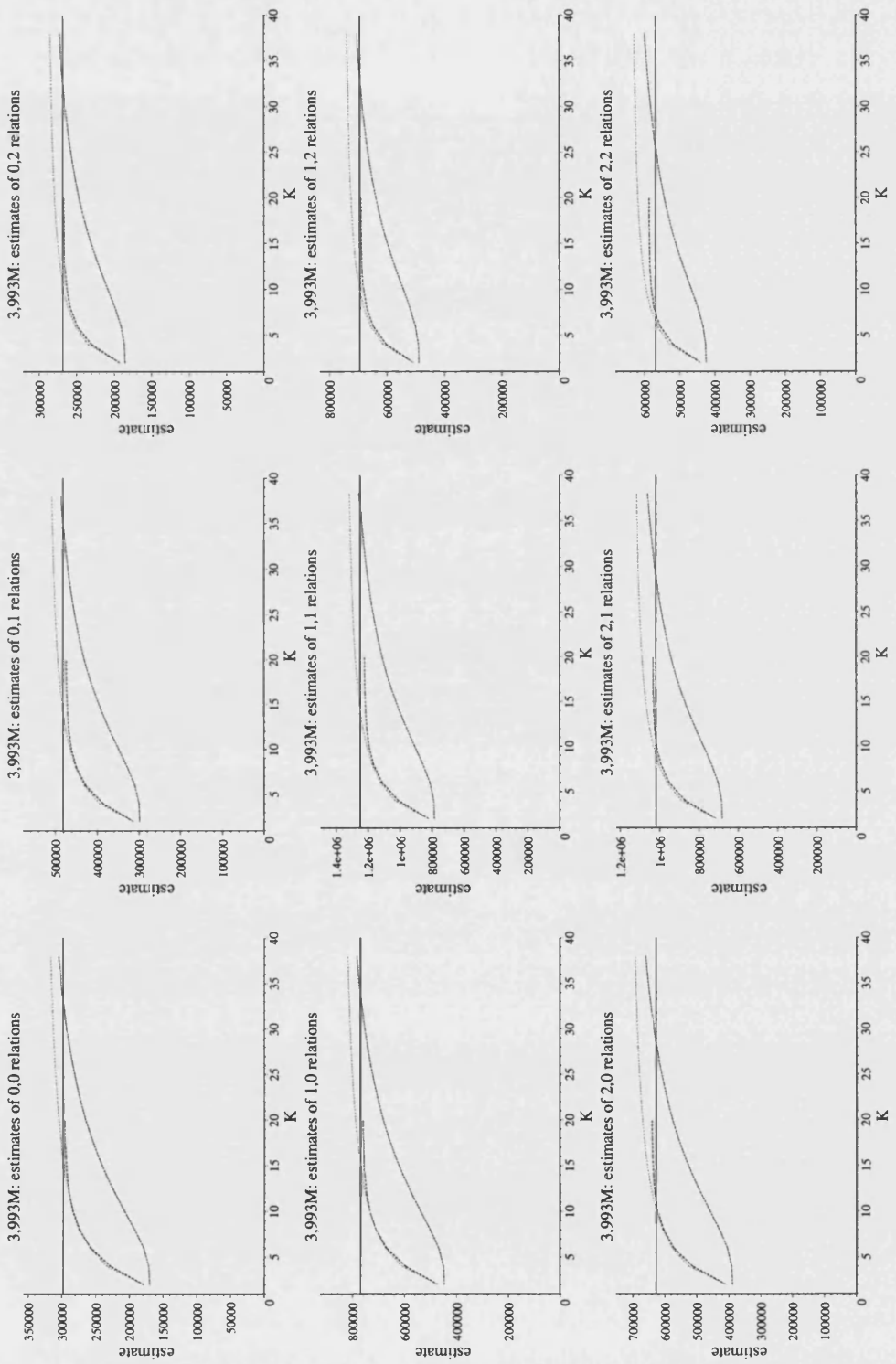
The equal splitting method produced an overestimate as K grows, although for small K the results are not unreasonable. Other examples in which we computed similar data for the equal splitting method showed the same overestimate. As the maximum value of K in this case is also extremely large there is no immediate way of either deciding what value of K to utilise or whether it is likely to have produced an overestimate of the type seen in the graphs below. It is noted that this overestimate may be the effect of accumulating error in the approximation or it may suggest that we are approximating a function that itself overestimates the yield in the range of interest.

Method 1 not only produces an overestimate for large values of K but in addition shows poor results for small K. Since there is no obvious manner in which to select the "right" value of K we turn to method 2. Method 2 looks the most promising of the three ways in which to split the sieve region. We will investigate this further by using the method to produce estimates for some additional factorisations.

4.5.1 Results

We calculate estimates using method 2 for factorisations which Cavallar tested the original method on in [14, 20]. In each case we use a maximal value of K,

Figure 4.6: $3,993M$: Comparison of estimates computed with the three methods



that is, the largest (even) value of K for which the interval $[1, b]$ splits into K subintervals of non-zero size. We present the results in a manner reminiscent of that of Cavallar to aid comparison. The factorisation names are those given by Cavallar, $x, y+$ denotes x^y+1, $2, 2hM$ denotes the Aurifeuillian factor $2^h+2^{\frac{h+1}{2}}+1$, $3, 3hM$ denotes $3^h+3^{\frac{h+1}{2}}+1$, $3, 3hL$ denotes $3^h-3^{\frac{h+1}{2}}+1$ and Fx are Fibonacci numbers. We recall factorisations presented by Cavallar in figure 4.7.

Figure 4.7: Polynomials used in factorisations

Number	$f_1(X)$	$f_2(X)$
$3,993M$	$3^{55}X-1$	X^6+3X^3+3
$3,999L$	$3^{55}X-1$	X^6-9X^3+27
$3,413+$	$X-3^{59}$	$X^6-X^5+X^4-X^3+X^2-X+1$
$3,427+$	$X-3^{61}$	$X^6-X^5+X^4-X^3+X^2-X+1$
$3,516+$	$3^{57}-1$	X^6+3X^3+9
$3,407+$	$3^{37}X-3^{74}-1$	$X^5-X^4-4X^3+3X^2+3X-1$
$F857$	$F171X-F172$	$X^5+5X^4+10X^2-5X+2$
$2,2130M$	$f_1(X)=X-5310903123331135610192$	
	$f_2(X)=6590263680X^5$	
	$-71058983292296X^4$	
	$+10126751094225398X^3$	
	$+349867764197537945X^2$	
	$-5404582433335517396810X$	
	$+2581409262310033997312415$	

We present results from our estimation technique in figures 4.8 and 4.9. The estimates are all within 12% of the actual results. In fact if we discount the factorisation $2,2130M$ the results are within 7% and usually closer. The factorisation $2,2130M$ required us to use a smaller value for K than suggested by the size of the a interval as the b interval was much smaller. It is possible that this has had an effect on the results — we might wish to use different values of K for the a interval and b interval in future work to determine if this is the cause. However, the results are much improved on the method with no splitting. The bounded value of K means we are able to select this parameter in a manner likely to enable us to produce good estimates and finally the method does not lead to any significant overestimates (as the equal splitting tends to) as K grows.

Finally we note that there is an increased computational cost attached to any method of splitting the region. If we split the region into K^2 subregions and then produce estimates for each of these we expect this to cost about K^2 times the

name	$3,993M$	$3,999L$	$3,413+$	$3,427+$	$3,516+$
degree f_1	1	1	1	1	1
degree f_2	6	6	6	6	6
A	1680000	2520000	3360000	4200000	3900000
B	1560000	1250000	2400000	3200000	1600000
B_1	4400000	8500000	11000000	14500000	8500000
B_2	11000000	10000000	13000000	17000000	10000000
L	60000000	80000000	100000000	100000000	90000000
$\alpha(F_1, B_1)$	0.569915	0.569915	0.569915	0.569915	0.569915
$\alpha(F_2, B_2)$	1.468072	1.429203	2.378699	2.377064	1.193893
K	20	20	20	22	20
full	297961/0.99	412555/0.97	502027/1.00	684987/0.99	408537/0.98
0, 1-partials	481365/0.98	873553/0.96	1047129/0.98	1205720/0.98	935790/0.97
0, 2-partials	268380/1.00	633695/0.97	759311/0.99	741788/0.99	742778/0.98
1, 0-partials	769170/0.99	806649/0.96	1008690/1.00	1194986/0.99	889398/0.97
1, 1-partials	1248973/0.98	1711506/0.95	2116479/0.98	2107447/0.98	2049612/0.96
1, 2-partials	694993/0.99	1245009/0.97	1532260/0.99	1299863/0.99	1628450/0.98
2, 0-partials	627188/1.02	500656/0.98	655488/1.03	686676/1.02	623474/1.00
2, 1-partials	1018741/1.01	1065195/0.97	1374882/1.01	1217910/1.01	1441725/0.99
2, 2-partials	568849/1.03	780025/0.99	1003843/1.02	752013/1.02	1148798/1.00

Figure 4.8: Degree 6 factorisation and estimation data

Figure 4.9: Degree 5 factorisation and estimation data

name	$3,407+$	$F857$	$2,2130M$
degree f_1	1	1	1
degree f_2	5	5	5
A	3600000	6000000	97200000
B	3000000	3050000	135000
B_1	13000000	11000000	4200000
B_2	10000000	13000000	16777215
L	100000000	100000000	100000000
$\alpha(F_1, B_1)$	0.569915	0.569915	0.569915
$\alpha(F_2, B_2)$	2.319329	1.002230	-5.915719
K	20	20	16
full	387672/0.93	393668/0.97	364736/0.92
0, 1-partials	737783/0.93	652752/0.96	812613/0.96
0, 2-partials	446398/0.94	336153/1.01	621128/1.12
1, 0-partials	944266/0.93	1095953/0.97	865394/0.91
1, 1-partials	1799413/0.93	1817042/0.96	1930024/0.95
1, 2-partials	1085377/0.94	937005/1.01	1471358/1.11
2, 0-partials	819125/0.95	1071958/1.01	574197/0.90
2, 1-partials	1565368/0.95	1779998/1.00	1279510/0.95
2, 2-partials	946628/0.96	916616/1.05	972034/1.12

cost of Cavallar's original method.

Further work

This investigation has illustrated that splitting the region can produce better estimates of yield however there are two overriding issues:

1. The method of splitting the region has a large impact on the results obtained and so we would prefer to take the polynomial values into account when splitting the region — as we can in the case of intervals.
2. We can provide no error estimates for the approximation.

As noted earlier it would be more natural to use a method that adapts the split into subregions to the particular polynomial and thus a method reminiscent of numerical quadrature would be an obvious direction for further work.

4.6 Summary

We have investigated improvements on the estimation technique due to Cavallar for estimating the quantity of full and partial relations that a classical sieve will produce.

We considered the reasons behind the significant underestimate on the non-linear side in the original method and following the suggestion of Cavallar we attempt to improve the estimate by splitting the sieve region. However we find that splitting the region into equal subregions, while improving the estimate does not solve the underlying problem and in fact results in an overestimate in the cases we considered in a process for which K can plausibly be taken to be very large.

We suggest a different method of splitting the sieve region — it is also quite simplistic but has certain attractive qualities: firstly, the estimates with increasing K appear to converge quite quickly toward the actual value; secondly, the method has a conclusion that is reached far more quickly than in the equal splitting case, that is, there is a reasonably small maximal value of K and finally the method easily out-performs estimates found without splitting the region in all cases we have tested though at an increased computational cost. However, due to the nature of the estimate it is not possible to confirm that the method will always produce a good estimate. Instead we provide evidence to support our claim.

We intend to use this method only to compare possible variants of the number field sieve, alongside other data and not to compare different parametrisations so we do not seek to verify the stability of the estimate with respect to variations in parametrisations of a single factorisation but leave this to further work. It is not suggested that this method should be used as a sole mechanism for examining a factorisation parametrisation or number field sieve variant without first establishing the precise abilities of the method either theoretically, which does not appear immediately possible, or through large scale tests for which we do not have the resources.

Finally we note a less naïve approach which would seem to be particularly suited to the problem at hand, this is left for further work.

Chapter 5

Characteristics of special number field sieve factorisations

The common thread running through all of the special cases is that the polynomials are chosen by hand using knowledge of some structure in the number to be factored. The polynomials chosen tend to have extremely small coefficients and hence the runtime of the algorithm is significantly reduced. In all cases we use a linear polynomial and a non-linear polynomial of small degree.

The original base-m general polynomial selection methods produce polynomials that one might argue are randomly selected. The coefficients are typically large in size, the Galois group, with probability approaching 1 will be the full symmetric group hence we have the minimum density of free relations. The root properties are generally poor.

When the polynomials are chosen by hand they often have certain characteristics that we would not usually expect to see in the general case. The most obvious characteristic is the extremely small size of the non-linear polynomial coefficients.

We also do not immediately assume that the Galois group of the polynomial is the full symmetric group — in fact the small size of the Galois group in various cases has been noted previously, by Huizing, for instance [49] since this guarantees us a more favourable density of free relations. We take this line of enquiry further and note that when d_1 is composite it is possible for the selected number field to

have a proper subfield.

We shall see that many of the special cases show different factor base structure on average than we might expect in the general case.

While we could encounter any number of specialist methods for selecting polynomials there are several that tend to be used on a regular basis and that produce polynomials of a specific form.

5.1 Polynomial selection methods for special cases

The original number field sieve [63] was designed to factor $n \in \mathbb{Z}$ where n or a small multiple of n is of the form $r^e - s$, $r > 1$, $|s|$ small positive integers, e large. Examples of numbers of this form include both Fermat and Cunningham numbers. We construct the number fields as follows: Choose $d_1 > 1$, let $k = \lceil e/d_1 \rceil$, $m = r^k$ then we have

$$\begin{aligned} f_1(X) &= X^{d_1} - sr^{kd_1 - e} \\ f_2(X) &= X - m. \end{aligned}$$

The most appropriate value for d_1 tends to infinity very slowly with the size of the number to be factored. At the time of writing it can be assumed that $d_1 \in \{4, 5, 6, 7\}$. The case $d_1 = 4$ is equivalent in general to MPQS but is occasionally better in special cases.

Other authors [72], alter this slightly by taking k to be the nearest multiple to e of d_1 and write $n = c_1(r^k)^{d_1} + c_2$ possibly by multiplying n by a small power of r.

Huizing and others [42, 49] note that we can use similar ideas when attempting to factor n of the form $n = c_1 r^{e_1} + c_2 s^{e_2}$, with r, s small positive integers, $e_1 \approx e_2$, $|c_1|$, $|c_2|$ small and finally, $\gcd(c_1 r, c_2 s) = 1$. Choose d_1 as before and set $k_1 = \lfloor e_1/d_1 \rfloor$,

$k_2 = \lfloor e_2/d_1 \rfloor$, $m = s^{k_2} r^{-k_1} \bmod n$ then we have

$$\begin{aligned} f_1(X) &= c_2 s^{e_2 - d_1 k_2} X^{d_1} + c_1 r^{e_1 - d_1 k_1}, \\ f_2(X) &= r^{k_1} X - s^{k_2}. \end{aligned}$$

If we have a number of the form $n = (r^e - 1)/(r^k - 1)$ we could use one of the above methods to produce small polynomials with which to use SNFS to factor $(r^e - 1)$ however this number could be significantly larger than n and require more sieving. It is also undesirable to use the general method on n since the polynomial would be significantly larger in this case. In some cases we are able to use the following method instead.

This method is particularly relevant when $e/k = 11$ or 13. Set $m = r^k$ and rewrite $n = (m^{e/k} - 1)/(m - 1) = g_1(m)$ where $g_1(X) = \sum_{i=0}^{e/k-1} X^i$. The polynomials $g_1(X)$ and $g_2(X) = X - m$ fulfil our criteria if the degree of g_1 is not large. If the degree of g_1 is too large and $(e/k - 1)/2$ is of a better size we can express $g_1(X)/X^{(e/k-1)/2}$ as a polynomial in $X + X^{-1}$. If $e/k = 11$ or 13 this will give us a polynomial $f_1(X)$ of degree 5 or 6 respectively. The same method will produce a reducible polynomial when $e/k = 9$. The polynomials that may be produced by this method are given in the table:

Degree	$f_1(X)$
5	$X^5 + X^4 - 4X^3 - 3X^2 + 3X + 1$
6	$X^6 + X^5 - 5X^4 - 4X^3 + 6X^2 + 3X - 1$

Table 5.1: Polynomials produced by this method.

We set $m = r^k + r^{-k} \bmod n$. Examples of factorisations using such polynomials can be found in [42].

Other methods of producing extremely small polynomials exist, algebraic and Aurifeuillian factorisations can often suggest useful polynomials that have small coefficients for instance see [11]. It is possible that any known structure in a number may allow us to select a "special" polynomial.

It is not possible to consider every method that may be conceived of for finding special polynomials. In the remainder we will focus on the above widely accepted

methods that lead to a polynomial f_1 of a specific form. Since we are discussing only the non-linear polynomial we will drop the subscripts.

5.2 Size properties

The first characteristic that we note is the overwhelmingly small size of the coefficients of the non-linear polynomials produced by these methods. In this respect these pairs represent an extreme. In fact this is why these polynomial pairs were originally seen as "good" — the difference in coefficient size is enough to produce a smaller asymptotic complexity.

Runtime of the original special case

If we consider the original method for producing polynomial pairs for integers of the form $n = r^e - s$ then we immediately see that the polynomial produced will always be monic with all other coefficients equal to 0 except for the constant term. The constant term itself is also bounded as follows:

$$|t| = |sr^{kd-e}| = |s|r^{kd-e} < |s|r^d$$

since $r > 1$ and

$$kd - e \geq d \implies k \geq \frac{e}{d} + 1$$

but $k = \lceil \frac{e}{d} \rceil < \frac{e}{d} + 1$.

Since r, $|s|$ are assumed to be small and $2 \leq d < 8$ (at the time of writing) we can assume that $|t|$ is also reasonably small (in comparison to the expected size of coefficients in the general case).

The coefficients of the linear polynomial are not as small, however the polynomial is also usually chosen to be monic. The constant term will be of size $\approx n^{1/d}$.

When we are able to select extremely good polynomials by hand we see this reflected in the asymptotic runtime. If we consider the original special number field sieve then we will be examining numbers of the form $|(a - bm)\mathbf{N}(a - b\alpha)| =$

$|(a - bm)(a^d - tb^d)|$ for smoothness. In this case the upper bound will be (if we assume $a_{\max} = -a_{\min}$ and $b_{\max} \leq a_{\max}$):

$$\begin{aligned} |(a - bm)(a^d - b^d t)| &\leq (a_{\max} + b_{\max} m)(a_{\max}^d + b_{\max}^d |t|) \\ &\leq a_{\max}^{d+1} + a_{\max}^{d+1}|t| + a_{\max}^{d+1} m + a_{\max}^{d+1} m|t| \\ &< 2|t| m a_{\max}^{d+1} \\ &\approx 2|t| n^{1/d} a_{\max}^{d+1} \end{aligned}$$

As noted in [63] we need only follow this change through the analysis of the general case. In this case the typical size of numbers that we test for B-smoothness becomes [63]

$$\exp\left((1/2 + o(1))\left(d^2 \log d + 2\log n^{1/d} + d\sqrt{(d\log d)^2 + 2\log n^{1/d} \log\log n^{1/d}}\right)\right).$$

In SNFS we then have the runtime

$$\exp\left((1 + o(1))\left(d\log d + \sqrt{(d\log d)^2 + 2\log n^{1/d} \log\log n^{1/d}}\right)\right).$$

It remains to note the optimal choice for d in the SNFS case:

$$d = \left(\frac{(3 + o(1))\log n}{2\log\log n}\right)^{1/3}$$

for $e \to \infty$ uniformly for r, s in a finite set. With this choice of d we find that the typical size of the numbers that we test for B-smoothness is then $L_n[2/3, (16/3)^{1/3}]$; which is $n^{o(1)}$ and the runtime is

$$L_n[1/3, (32/9)^{1/3}].$$

Other special forms

In the case of the method for integers of the form $n = c_1 r^{e_1} + c_2 s^{e_2}$ we have a polynomial of the form $f(X) = c_2 s^{e_2 - dk_2} X^d + c_1 r^{e_1 - dk_1}$ and we will find that the coefficients are bounded by the same argument as above.

The non-linear polynomial in the final method detailed above clearly has extremely small height (this can be seen by examination of the specific polynomials).

While there are examples of special cases where several different polynomial pairs have been produced and the decision of which to use was based on other properties it is often the case that the smallest polynomial is used. A key question is whether this is the correct course of action. The algorithms produced by Murphy for polynomial selection in the general case, that have had such positive results take into account *both size and root properties* however he assumes that the underlying basic method of finding polynomials will effectively produce polynomials that have been selected at random. What other properties do SNFS polynomials have?

5.3 Root properties

In chapter 3 we outlined the function $\alpha(F)$ which aimed to capture the difference in probability between a polynomial value $F(a, b)$ and a random number, of the same size, being smooth. Ideally we would prefer that the F values acted like random integers a great deal smaller and hence we would like this value to be as negative as possible.

We note that $\alpha(F) > 0$ would correspond to the values $F(X, Y)$ being less likely to be smooth over the primes $p \leq B$ than random integers of the same size. This situation can occur when we have very few first degree prime ideals of small norm but also when, for a high proportion of the primes $p \leq B$, we have no first degree prime ideals of norm p.

In order to consider the root properties in the special cases, the possibly of finding subfield structure and the density of free relations we look at the Galois group of the polynomials produced for many of the SNFS factorisations.

5.3.1 The Galois group in special cases

In the general case it is assumed that the Galois group will be S_d, the full symmetric group, the set of permutations of d elements with size $d!$. In this case we have the minimum quantity of free relations and cannot have subfields. It is already known that special case polynomials may have smaller groups and we would like to investigate this in more detail.

We would be interested to know the Galois group of these fields so that we may estimate the density of free relations. This can also be useful when determining information on any subfields. Since f is irreducible we know that $\mathrm{Gal}(f)$ the Galois group of the splitting field of f is a transitive permutation group.

The transitive permutation groups of degree up to 15 are summarised in [25, section 2.2 but see also 1.2] and I use the naming system of that paper below.

In the original polynomial construction method we will always produce a monic non-linear binomial. The variations of this method will produce a non-linear binomial which may not be monic hence we will work with $f(X) = cX^d - t$, $c, t \in \mathbb{Z}$, $c, t \neq 0$. We may assume that f is irreducible over $\mathbb{Q}$. In this case we are able to say a great deal about the possible Galois groups.

Let α be a root of f, and define $K = \mathbb{Q}(\alpha)$. The minimum polynomial of α will be $\hat{f}(X) = X^d - t/c \in \mathbb{Q}[X]$ with $t/c \in \mathbb{Q}$, $t/c \neq 0$. Let S be the splitting field of $\hat{f}$ (and hence of f) and ζ be any primitive d^{th} root of unity, then the other roots of $\hat{f}$ have the form $\zeta^i\alpha$ for $1 \leq i \leq d-1$. Hence the splitting field is $S = K(\zeta) = \mathbb{Q}(\alpha, \zeta)$ and $S = K$ if and only if $\zeta \in K$. Now we have three cases:

1. If $S = K$ then we have $[S : \mathbb{Q}] = d$ and hence the size of the Galois group is d.
2. If $S = K(\zeta) = \mathbb{Q}(\alpha, \zeta)$ and the dth cyclotomic polynomial is irreducible over K then the minimum polynomial of ζ is the d^{th} cyclotomic polynomial which has degree $\phi(d)$ where ϕ is Euler's phi (or totient) function. Then we have $[\mathbb{Q}(\alpha) : \mathbb{Q}] = d$ and $[\mathbb{Q}(\alpha, \zeta) : \mathbb{Q}(\alpha)] = \phi(d)$ and so $[\mathbb{Q}(\alpha, \zeta) : \mathbb{Q}] = [\mathbb{Q}(\alpha, \zeta) : \mathbb{Q}(\alpha)][\mathbb{Q}(\alpha) : \mathbb{Q}] = \phi(d)d$.
3. If the dth cyclotomic polynomial factors over K but we are not in case 1 (it

does not factor completely) then roots are discussed in Lang [59, theorem 9.4 and remarks], (when $d \in \{4, 6\}$ this case cannot occur as the polynomial in question is a quadratic).

The elements of the Galois group are found by considering the distinct $\mathbb{Q}$-automorphisms of S. Any $\mathbb{Q}$-automorphism of S permutes roots of the defining minimum polynomials, that is, we consider all distinct combinations of

$$\begin{aligned} &\varsigma_i : \alpha \mapsto \alpha\zeta^i, \quad \varsigma_i : \zeta \mapsto \zeta, && 0 \le i \le d-1, \\ &\tau_j : \zeta \mapsto \zeta^j, \quad \tau_j : \alpha \mapsto \alpha, && 1 \le j \le d-1,\ \gcd(j,d) = 1. \end{aligned}$$

We are primarily interested in f of degrees < 7:

- $d = 2$: There is only one possible Galois group, the full symmetric group S_2.
- $d = 3$: The above results give $|\operatorname{Gal}(\hat{f})| = 3$ or $|\operatorname{Gal}(\hat{f})| = 6$.

  A 3^{rd} primitive root of 1 is in $\mathbb{C}$ and not in $\mathbb{R}$. However, a function of the form $X^3 - t/c$ will always have a real root and hence a real embedding. We can immediately deduce that we have no primitive 3^{rd} primitive root of 1 in K and hence the only possibility is $\operatorname{Gal}(\hat{f}) = S_3$.
- $d = 4$: i is a primitive 4^{th} root of 1, if $i \notin K$ then the Galois group has order 8 and is the dihedral group $D(4)$.

  If $i \in K$ then the Galois group has order 4 and there are two possibilities: $C(4)$ and $E(4)$. If we consider the elements of the Galois group itself, we find that the group is generated by $\alpha \mapsto -\alpha$ and $i \mapsto -i$ and hence we must have $\operatorname{Gal}(\hat{f}) = E(4)$.
- $d = 5$: The Galois group is of order 5 or 20. The only possible Galois group of 5 elements is $C(5)$ and the only possible Galois group of 20 elements is $F(20)$.

  By a similar argument to that given in the case $d = 3$ above there can be no primitive 5^{th} root of 1 in K and we can immediately deduce $\operatorname{Gal}(\hat{f}) = F(20)$.
- $d = 6$: $\operatorname{Gal}(\hat{f})$ is of order 6 or 12. If there is no primitive 6^{th} root of 1 in K then $\operatorname{Gal}(\hat{f}) = D(6)$ or $\operatorname{Gal}(\hat{f}) = A_4(6)$. If we consider the form of

the Galois group given above we can immediately conclude that it is of the form $S(3) \times C(2)$ and hence is the group $D(6)$.

If there is a primitive 6^{th} root of 1, say ζ, in K then either $\text{Gal}(\hat{f}) = C(6)$ or $\text{Gal}(\hat{f}) = D_6(6)$. The Galois group is generated by ς_1^2, and τ_5 and hence we must have $\text{Gal}(\hat{f}) = D_6(6)$.

We could continue in this way noting the specific groups. In general we note only that the size of the Galois group must be less than or equal to $d\phi(d)$ and hence for $d > 3$ the Galois group is always strictly smaller than the full symmetric group in all cases.

The final method given above that is used to produce polynomials can give rise either to a polynomial of the form $f(X) = \sum_{i=0}^{d} X^i$ (which we may use if $d = 4, 6$) or, if $d = 12$, we can produce the polynomial of degree 6 given in table 5.1. In all of these cases the Galois group of the polynomial will be $C(d)$ (this may be checked using, for instance, KASH [32]). In the case where $f(X) = \sum_{i=0}^{d} X^i$, $d = 4, 6$ we can see this since the polynomial is a cyclotomic polynomial. In both cases the roots of unity will be present in K.

Other "special" polynomials that have been used in SNFS factorisations also have small Galois group in all cases which we encountered (see appendix A for the more prominent examples). In the most part the groups are $C(4)$, $E(4)$, $D(4)$, $C(5)$, $F(20)$, $C(6)$, $D_6(6)$ or $D(6)$ although in the case of $d = 6$ we have also encountered the group $F_{18}(6)$.

5.3.2 Factor base structure

Another characteristic that sets special cases apart is the structure of the factor base, where there are some properties that differ from the general case.

Since special case polynomials are not chosen with particular regard to the root properties present we would be interested to know something of the average root properties that they have. This is difficult to quantify for general d or even for non-monic polynomials so we will consider the most pertinent degrees 4, 5 and 6 in detail. For this section we will assume that f is a monic polynomial. If f is not

monic then the primes that divide the leading coefficient will not be encompassed by the theory discussed below, however this will be a finite set of primes. If these are small primes then the effect may be significant — as demonstrated by Murphy's schema, however, when selecting SNFS polynomials we do not have an immediate method of forcing a large quantity of small projective roots and so cannot control this effect to our advantage.

In the general case the degrees 4, 5 and 6 do produce slightly different general factor base structures but these are not significant and Murphy discounts them. However, his argument is based on the assumption that the polynomial used is selected at random (and hence will have Galois group S_d with probability approaching 1) and that his aim is to select polynomials with better than average properties.

Neither of these assumptions are valid in the main special cases of the number field sieve so we will reconsider the implications. We will compare the factor base structure in the two cases and then consider whether the values of $\alpha(F)$ and $\mathbb{E}(F)$ are still relevant when comparing special case polynomials.

In [76, chapter 3] Murphy considers whether the choice of d will influence the root properties to any significant extent. We revisit this, contrasting the special and general cases.

5.3.3 The factor base structure when f is a randomly selected polynomial

Theorem 2 *[45] Let* $f(X) = X^d + c_{d-1}X^{d-1} + \cdots + c_0 \in \mathbb{Z}[X]$ *be a monic polynomial of degree d with Galois group* $\mathrm{Gal}(f)$ *a subgroup of* S_d. *Let* $N_d(h)$ *be the number of such polynomials with* $\max(|c_{d-1}|, \ldots, |c_0|) \leq h$ *for which* $\mathrm{Gal}(f) < S_d$. *Then*

$$N_d(h) << h^{d-1/2} \log^{1-\epsilon} h$$

where $\epsilon > 0$ *is dependent on d.*

Informally this result states that most (asymptotically density 1, independent of d) monic polynomials $f(X) \in \mathbb{Z}[X]$ of degree d have Galois group isomorphic

to the full symmetric group. Thus if we select such a polynomial at random it has Galois group $\mathrm{Gal}(f) < S_d$ with probability approaching 0. While the base-m method is not random, no attempt is made to select polynomials with strictly smaller Galois groups. It seems likely that most polynomials produced with this method will have $\mathrm{Gal}(f) \cong S_d$ and Murphy assumes that this is the case. We have seen that known methods taking advantage of structure usually produce polynomials with Galois groups strictly smaller than the full symmetric group.

Definition 6 *The cycle shape of a permutation is the multiset of the lengths of its cycles when the permutation is written as a product of disjoint cycles. We say that a permutation has k fixed points if it has k disjoint cycles of length* 1.

A consequence of the Chebotarev Density Theorem is the following [55]:

Proposition 3 *The degrees of the irreducible factors of f modulo p, p prime and f square free, are the same as the cycle shape of an element of the Galois group of f over $\mathbb{Q}$. Further, the proportion of $p \leq B$ giving rise to a certain shape tends to the proportion of elements of the Galois group having that shape as $B \rightarrow \infty$.*

The formula to calculate the number of permutations of d elements that have k fixed points is given in [87]

$$FP(d,k) := \frac{d!}{k!} \sum_{i=0}^{d-k} \frac{(-1)^i}{i!}.$$

We can see that the proportion of the $d!$ possible permutations of d elements with k fixed points is:

$$\frac{FP(d,k)}{d!} = \frac{d!}{k!d!} \sum_{i=0}^{d-k} \frac{(-1)^i}{i!} = \frac{1}{k!} \sum_{i=0}^{d-k} \frac{(-1)^i}{i!}.$$

We will identify the proportion of integer primes below B, as $B \rightarrow \infty$, that do not appear in the algebraic factor base and that can therefore never divide $\mathbf{N}(a - b\alpha)$, this is the case $k = 0$ and such primes will contribute positive values to $\alpha(F)$. We also consider the case $k = 1$, the proportion that are present only

once (which contribute 0 to $\alpha(F)$), and finally those present multiple times, which contribute negatively to $\alpha(F)$. We will state results only for those primes that do not divide the discriminant of f, since those that do will not follow a general pattern. Obviously only a finite number of primes can divide the discriminant (which for an SNFS polynomial will be small).

$k = 0$:

We identify the proportion of integer primes $p < B$, as $B \to \infty$, for which we have no first degree prime ideals of norm p. This is equal to the proportion of the $d!$ permutations contained in S_d which have no cycle of length one:

$$\frac{1}{0!}\sum_{i=0}^{d-0}\frac{(-1)^i}{i!} = \sum_{i=0}^{d}\frac{(-1)^i}{i!} = \frac{1}{1} - \frac{1}{1} + \frac{1}{2} - \frac{1}{6} + \frac{1}{24} - \ldots + \frac{(-1)^d}{d!}.$$

When $d = 2$ we have the proportion $1/2$, when $d = 3$ we have the proportion $1/3$. For $d > 3$ we truncate the sum after the 5th term to produce an upper bound of $3/8 \approx 0.375$ and after the 6th term to produce a lower bound of $11/30 \approx 0.366$ (in fact the well known limit as $d \to \infty$ is $1/e$).

$k = 1$:

We now note the proportion of integer primes $p < B$, as $B \to \infty$, for which we have one first degree prime ideal of norm p. This is equal to the proportion of the $d!$ permutations contained in S_d which have exactly one cycle of length one:

$$\frac{1}{1!}\sum_{i=0}^{d-1}\frac{(-1)^i}{i!} = \sum_{i=0}^{d-1}\frac{(-1)^i}{i!}.$$

The first few terms of this are the same as in the case $k = 0$.

When $d = 2$ we have no cycles of length exactly one, when $d = 3$ we have the proportion $1/2$. For $d > 3$ we truncate the sum after the 4th term to find a lower bound of $1/3 \approx 0.33$ and after the 5th term to find an upper bound of $3/8 \approx 0.375$.

$k > 1$:

Finally we calculate the proportion of integer primes $p < B$, as $B \to \infty$ for which we have strictly more than one first degree prime ideal of norm p. This is equal to the proportion of the $d!$ permutations contained in S_d which have more than one cycle of length one and this is 1 minus the total proportion of permutations with no cycles or exactly 1 cycle.

When $d = 2$ we have $1 - \frac{1}{2} = \frac{1}{2}$. These will have exactly two cycles of length one. When $d = 3$ we have $1 - (\frac{1}{2} + \frac{1}{3}) = \frac{1}{6} \approx 0.166$ of cycle lengths two or three. When $d > 3$ we have an upper bound of $1 - (\frac{11}{30} + \frac{1}{3}) = \frac{3}{10} \approx 0.3$ and a lower bound of $1 - (\frac{3}{8} + \frac{3}{8}) = \frac{1}{4} \approx 0.25$.

We conclude, as Murphy did, that in the general case there is little to choose between various d, $d > 3$ in terms of the average root properties produced.

5.3.4 Factor base structure in special cases

When working in complete generality, it can be difficult to compare factor bases obtained for number fields with Galois group the full symmetric group with typical SNFS factor bases. We consider the three most pertinent degrees.

$d = 4$

From figure 5.1 we observe that for SNFS factor bases we have a significantly high proportion of integer primes for which there is no first degree prime of that norm. In particular, when f is chosen such that the Galois group is the cyclic group $C(4)$ or the Klein group $E(4)$ these cases are expected to have 3/4 of integer primes below B, as $B \to \infty$, not present in our factor base.

In the case of Galois group S_4 we have $k > 1$ on average 29% of the time, in the special cases $C(4)$ and $E(4)$ on average 25% of the time. The case of $D(4)$ stands out with $k > 1$ on average 38% of the time.

However, the percentage of primes that, on average, will not be present in the factor base at all could be cause for concern, unless the distribution for the particular polynomial favours the small primes, we may have to work harder to find smooth polynomial values than we would when working with random integers of the same size.

$d = 5$

If we consider figure 5.2 which corresponds to $d = 5$, we see the situation that occurred in many of the early factorisations using the special and general number field sieves (the Galois group $F(20)$ occurs for all polynomials produced by the original SNFS polynomial selection method with $d = 5$). In this case 4/5 of the primes $p \leq B$, as $B \rightarrow \infty$, are present.

However, we have only 5% of the primes with $k > 1$, on average. This is a disappointingly low figure — it suggests that while we are less likely to encounter very positive $\alpha(F)$ values (since 80% of primes are present at least once) it will also be difficult to attain particularly negative $\alpha(F)$ values as such a low proportion of primes will provide a negative contribution.

$d = 6$

If we consider figure 5.3, we conclude that in the cases where the Galois group is of size 6, we will have only 1/6 of the primes $p \leq B$ present, as $B \rightarrow \infty$. Unless a substantial quantity of these are the primes of reasonably small norm, the factor

Figure 5.1: For each group of interest: the quantity of each possible cycle shape present and the proportion of rational primes less than B which correspond to $0, 1, 2$ or 4 first degree primes as $B \longrightarrow \infty$

			2		3		No. first degree primes			
G	$\|G\|$	1^4	1^2	2^2	1	4	4	2	1	0
$C(4)$	4	1		1		2	1/4	0	0	3/4
$E(4)$	4	1		3			1/4	0	0	3/4
$D(4)$	8	1	2	3		2	1/8	1/4	0	5/8
S_4	24	1	6	3	8	6	1/24	1/4	1/3	3/8

base could be significantly worse than one that we would expect to encounter when the Galois group of f is S_d. While the situation is improved when we have the Galois group $D(6)$, we are missing almost 2/3 of the integer primes.

On the positive side all those present are present multiple times and so will contribute negatively to $\alpha(F)$ nonetheless, unless many small primes occur 2 or 6 times, we may expect to have to work harder to produce B-smooth values.

5.3.5 Root properties, $\alpha(F)$ and $\mathbb{E}(F)$

Asymptotically, the average number of pairs (p, r) for a given integer p is 1 if the smoothness bound is large enough [58]. However the particular structure of the factor base differs in more subtle ways. While it is possible that a polynomial leading to a factor base with a low proportion of integer primes is better than a polynomial with Galois group S_d, it is by no means certain. To improve on such a polynomial, either more integer primes must be accounted for, or the particular polynomial must have many roots modulo small primes.

Without aiming to manipulate the root properties of the polynomials used we may find that we are using polynomials with positive $\alpha(F)$ values. In fact, of the SNFS factorisations we have encountered we have found that the vast majority have positive values of $\alpha(F)$, details of the characteristics of a collection of the more prominent SNFS factorisations are contained in appendix A.

Since the polynomials used in special cases come from the structure of the number to be factored the level of control over the leading coefficient is low and hence we are less able to manipulate the distribution of the roots to ensure a negative

Figure 5.2: For each group of interest: the quantity of each possible cycle shape present and the proportion of rational primes less than B which correspond to $0, 1, 2, 3$ or 5 first degree primes as $B \longrightarrow \infty$

			2	2^2	3	3	4		No. first degree primes				
G	$\lvert G\rvert$	1^5	1^3	1	1^2	2	1	5	5	3	2	1	0
$C(5)$	5	1						4	1/5	0	0	0	4/5
$F(20)$	20	1		5			10	4	1/20	0	0	3/4	1/5
S_5	120	1	10	15	20	20	30	24	1/120	1/12	1/6	7/24	11/30

Figure 5.3: For each group of interest: the quantity of each possible cycle shape present and the proportion of rational primes less than B which correspond to $0, 1, 2, 3, 4$ or 6 first degree primes as $B \longrightarrow \infty$

							3					
			2	2^2		3	2		4	4	5	
G	$\|G\|$	1^6	1^4	1^2	2^3	1^3	1	3^2	1^2	2	1	6
$C(6)$	6	1			1			2				2
$D_6(6)$	6	1			3			2				
$D(6)$	12	1		3	4			2				2
S_6	720	1	15	45	15	40	120	40	90	90	144	120

No. first degree primes					
6	4	3	2	1	0
1/6	0	0	0	0	5/6
1/6	0	0	0	0	5/6
1/12	0	0	1/4	0	2/3
1/720	1/48	1/18	3/16	11/30	53/144

$\alpha(F)$ value. This means that we are likely to be more dependent on the average cases which we have just detailed. As we have seen this is highly variable and dependent not only on the degree but also on the polynomial form (two polynomials which share the same degree but have different Galois groups can have manifestly different average factor base structures).

We might wish to consider whether we can manipulate the distribution without destroying other positive properties — such as the extremely small size or beneficial Galois group. We note that this cannot be achieved by translations (which have no effect on root properties) or rotations (in general these will produce a different Galois group) as used by Murphy. On the other hand, it is possible to create isomorphic number fields which have different projective roots.

Nevertheless the options aren't wide open — we can alter things in a few ways but we will have less choice and hence the range of possible $\alpha(F)$ values (for polynomials for an integer n which share the same degree) is limited.

In addition we must consider that creating a polynomial that has a leading coefficient divisible by powers of many small primes will alter the size properties. We may wish to think before doing so since our reduced asymptotic runtime is

based in the extremely small size that the special cases provide.

It would seem that we may be sacrificing good root properties in order to achieve this small size — and seemingly for good reasons. This leads us to the conclusion that knowledge of $\alpha(F)$ values alone are of little or no benefit to us in special cases, in order to compare these with each other and the general case we must consider a measure such as $\mathbb{E}(F)$ which takes into account size and root properties.

Hence we are dealing with polynomials with significantly different characteristics from those that Murphy assumed.

5.4 Subfield structure in special cases

Notation

We will assume that we have polynomials f_1, f_2 and that f_1 is a non-linear polynomial of composite degree. For this section we will write $f = f_1$. We select a particular root of f denoted by α and define a number field $K = \mathbb{Q}(\alpha)$. We are interested in the situation where there is some field $L = \mathbb{Q}(\beta)$ such that $K \supset L \supset \mathbb{Q}$. Each such field that we define will be described by a polynomial $g(X) \in \mathbb{Z}[X]$ with root β and the embedding of β into K. The embedding will be given by a polynomial $h(X) \in \mathbb{Q}[X]$ with $h(\alpha) = \beta$. We have the following, for instance from [53],

Lemma 3 *A subfield L of K has a representation by a pair (g, h) with $g(h) \equiv 0 \pmod{f\mathbb{Z}[X]}$ and any such pair (g, h) describes a subfield of K.*

We note that h is not necessarily integral because $\mathbb{Z}[\alpha]$ is in general not a maximal order.

If the field has prime degree then there are no subfields hence we will only consider degree 4 and degree 6 (as the only composite degrees of a practical size at the time of writing). When $d = 4$, in the cases we are considering there will always be at least one quadratic subfield (as the Galois groups are of order 4, are 2-groups

and hence have an index-2 subgroup). We specify this subfield and check for other degree 2 subfields. In the second case we may have subfields of both degree 2 and degree 3.

There are a variety of algorithms for determining subfields in the most general cases however we will have far more information about the main field and the subfields to require these in the SNFS cases. If we wish to move to more general cases than the methods examined in [36, 50, 53, 60] will become more important.

Once we know the Galois group of the defining irreducible polynomial we are then able to find any subfields of the field $\mathbb{Q}(\alpha)$. We will look at the cases $d = 4$ and $d = 6$.

$d = 4$

Firstly we will assume that $d = 4$, $f(X) = cX^4 - t$ and the Galois group is $D(4)$ the dihedral group of 8 elements with generators

$$\varsigma = \varsigma_1, \quad \varsigma(\alpha) = i\alpha, \quad \varsigma(i) = i$$
$$\tau = \tau_3, \quad \tau(\alpha) = \alpha, \quad \tau(i) = -i$$

subgroup	id	ς	ς^2	ς^3	τ	$\varsigma\tau$	$\varsigma^2\tau$	$\varsigma^3\tau$	#
Triv.	•								1
$C(2)$	•		•						2
$C(4)$	•	•	•	•					4
$C(2)$	•				•				2
$C(2)$	•					•			2
$C(2)$	•						•		2
$C(2)$	•							•	2
$C(2) + C(2)$	•		•		•		•		4
$C(2) + C(2)$	•		•			•		•	4
$D(4)$	•	•	•	•	•	•	•	•	8

Figure 5.4: Degree 4, $D(4)$; subgroups

The subgroup inclusions can be readily produced from the figure 5.4 and under the Galois correspondence we obtain the intermediate fields. The correspondence reverses inclusions and so we have figure 5.5.

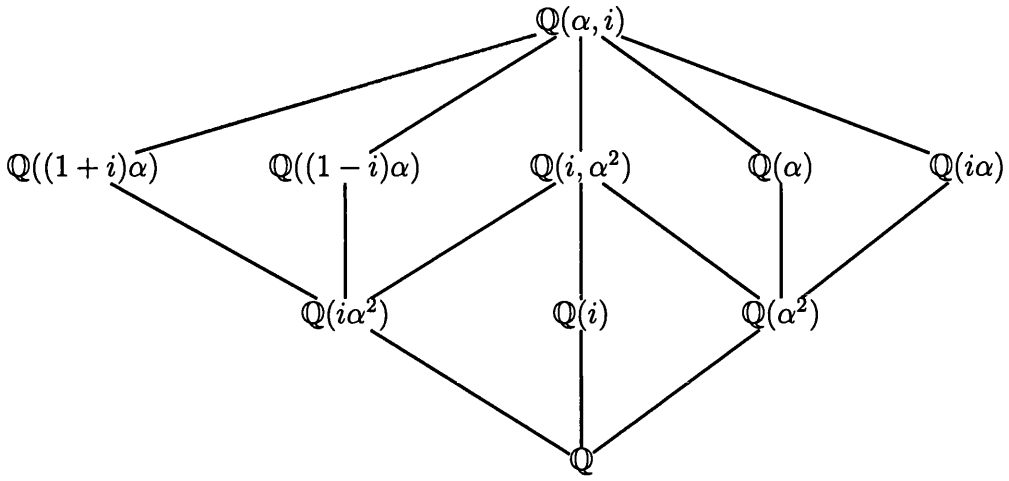


Figure 5.5: Degree 4, $D(4)$; subfields

It can be observed that $\mathbb{Q}(i)$, $\mathbb{Q}(\alpha^2)$ and $\mathbb{Q}(i\alpha^2)$ are the subfields of $\mathbb{Q}(\alpha, i)$ of degree 2. To find the others we note that any element of the splitting field can be written as

$$\gamma = a_0 + a_1\alpha + a_2\alpha^2 + a_3\alpha^3 + a_4 i + a_5 i\alpha + a_6 i\alpha^2 + a_7 i\alpha^3$$

and we then consider the action of the non-trivial group element (which should fix γ). We conclude that in this case we will always have exactly one subfield of $\mathbb{Q}(\alpha)$ of degree 2, the field is $\mathbb{Q}(\alpha^2)$. We can then see that there are two possible embeddings $h(X) = \pm X^2$. The embedding that is used depends on the particular roots of f and g that are adjoined to $\mathbb{Q}$.

When $d = 4$ and K is the splitting field (we have the Galois group $E(4)$) we work in a similar manner. We note that the situation can only arise when for $f = cX^4 - t$, $t < 0$, $|t| = s_1^2$, $c > 0$, $c = s_2^2$ and $s_i \in \mathbb{Z}$ (this ensures that $i \in K$). Letting ς and τ be as defined above the subgroups are shown in figure 5.6 and under the Galois correspondence we have figure 5.7.

We note that there are three subfields in this case with defining polynomials $g_1(X) = s_2X^2 + 2s_1$, $g_2(X) = cX^2 - t$, $g_3(X) = s_2X^2 - 2s_1$ with embeddings $h_1(X) = \pm(s_2/s_1X^3 + X)$, $h_2(X) = \pm X^2$ and $h_3(X) = \pm(-s_2/s_1X^3 + X)$

subgroup	id	ς^2	τ	$\varsigma^2\tau$	#
Triv.	•				1
$C(2)$	•	•			2
$C(2)$	•		•		2
$C(2)$	•			•	2
$E(4)$	•	•	•	•	4

Figure 5.6: Degree 4, $E(4)$; subgroups

respectively. These can be verified by application of lemma 3.

When $d = 4$ the Galois group is $E(4)$ or $D(4)$. In both cases we have at least one quadratic subfield which is defined by $(cX^2 - t, \pm X^2)$, the particular embedding depends on which roots are used. That this is indeed a subfield can be verified using the lemma above. In the case $E(4)$ there are two other quadratic subfields which are not isomorphic to this one but have $h \notin \mathbb{Z}[X]$ and hence we will not use these. We also mentioned the possibility of using a polynomial with $C(4)$ as the Galois Group. In this case, we will find that we have one subfield with integral embedding however, the embedding can be of a more complex nature. Such cases would need to be considered individually.

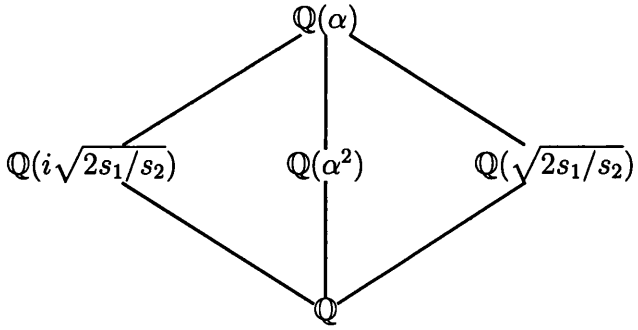


Figure 5.7: Degree 4, $E(4)$; subfields

$d = 6$

Assume $f(X) = cX^6 - t$ with Galois group $D(6)$. The subgroups of $D(6)$ can be found in a similar way to those of $D(4)$ above. Let ς and τ generate the group:

$$\varsigma = \varsigma_1, \ \varsigma(\alpha) = \alpha\zeta, \ \varsigma(\zeta) = \zeta$$
$$\tau = \tau_5, \ \tau(\zeta) = \zeta^5 \ \tau(\alpha) = \alpha$$

where ζ is the primitive 6^{th} root of 1 and is $e^{2i\pi/6} = (1 + \sqrt{3}i)/2$.

subgroup	id	ς	ς^2	ς^3	ς^4	ς^5	τ	$\varsigma\tau$	$\varsigma^2\tau$	$\varsigma^3\tau$	$\varsigma^4\tau$	$\varsigma^5\tau$	#
Triv.	•												1
$C(2)$	•			•									2
$C(3)$	•		•		•								4
$C(6)$	•	•	•	•	•	•							6
$C(2)$	•						•						2
$C(2)$	•							•					2
$C(2)$	•								•				2
$C(2)$	•									•			2
$C(2)$	•										•		2
$C(2)$	•											•	2
$C(2) + C(2)$	•			•			•			•			4
$C(2) + C(2)$	•			•				•			•		4
$C(2) + C(2)$	•			•					•			•	4
S_3	•		•		•		•		•		•		6
S_3	•		•		•			•		•		•	6
$D(4)$	•	•	•	•	•	•	•	•	•	•	•	•	12

Figure 5.8: Degree 6, $D(6)$; subgroups

The inclusions for degree 6 are far more complex and can be readily found using the figure 5.8. We consider only the inclusions that are interesting, these are noted in figure 5.9.

In this case we have two fields with defining polynomials $g_1(X) = cX^3 - t$ and $g_2(X) = cX^2 - t$ with embeddings $h_1(X) = X^2$ and $h_2(X) = \pm X^3$ respectively.

When K is the splitting field we note that we are always in the situation where $f(X) = cX^6 - t$, $t < 0$, $|t| = 3s_1^2$, $c > 0$, $c = s_2^2$, and $s_i \in \mathbb{Z}$ (this ensures that a

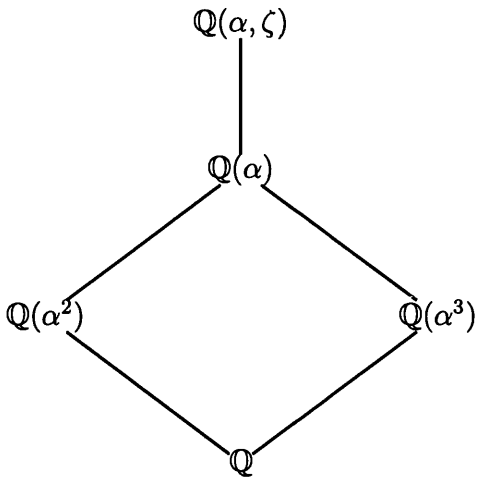


Figure 5.9: Degree 6, $D(6)$; subfields

primitive 6^{th} root of 1 is in K). Let ς and τ be as defined above and hence we have the subgroups:

subgroup	id	ς^2	ς^4	τ	$\varsigma^2\tau$	$\varsigma^4\tau$	#
Triv.	•						1
$C(3)$	•	•	•				3
$C(2)$	•			•			2
$C(2)$	•				•		2
$C(2)$	•					•	2
$D_6(6)$	•	•	•	•	•	•	6

Figure 5.10: Degree 6, $D_6(6)$; subgroups

The subgroup inclusions can be readily seen from the figure 5.10 and under the Galois correspondence we have figure 5.11.

In this case the defining polynomials are $g_1(X) = cX^2 - t$ and $g_i(X) = cX^3 - t$ for $i \in 2,3,4$. The latter three subfields, $\mathbb{Q}(\alpha_i^2)$, $i = 2,3,4$ are isomorphic but not equal but this can only be seen by noting that the embeddings are $h_1(X) = \pm X^3$, $h_2(X) = X^2$, $h_3(X) = -\frac{s_2}{2s_1}X^5 - \frac{1}{2}X^2$ and $h_4(X) = \frac{s_2}{2s_1}X^5 - \frac{1}{2}X^2$ respectively.

For larger values of d working in this way would become extremely complex

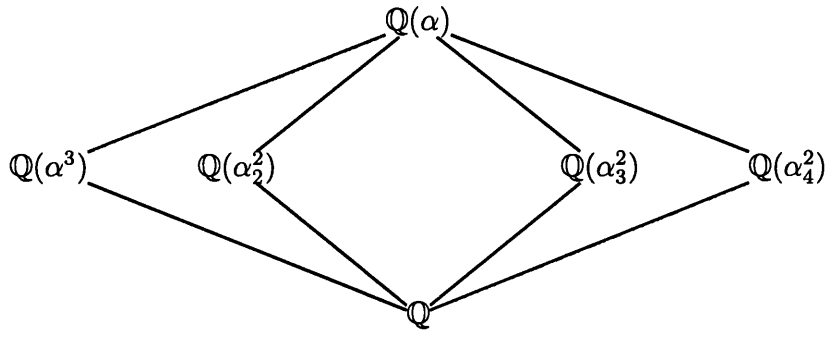


Figure 5.11: Degree 6, $D_6(6)$; subfields

and unnecessary if all that is required is a subfield equation and embedding. Algorithms to achieve this for particular fields can be found in [53].

If $d = 6$ the possible groups are $C(6)$, $D_6(6)$ and D_6. We will have both a quadratic and a cubic subfield. In the latter two cases the two obvious subfields are those in which we are interested. These are defined by $(cX^2 - t, \pm X^3)$ and $(cX^3 - t, X^2)$, again any additional subfields have $h \notin \mathbb{Z}[X]$. In the case $C(6)$ we again have two integral subfields with degree 2 and 3 however the embeddings are of a more complex nature.

For other forms of special polynomial we must work on a case by case basis, we are by no means assured subfields in all situations but the possibility arises frequently enough to be of interest.

5.5 Summary

We have explored various characteristics which are present in special cases of the number field sieve and have considered some of the possible Galois groups which may be found noting that these differ from what we might expect in general. Taking this argument further we have seen that there is subfield structure in some of the main special cases which use a field of composite degree. The size and root properties which may be expected have also been considered. Here we

see a significant departure from the general case, it is well known that the special cases are extremely small — in fact it is for this reason that they were originally selected. However, Murphy and others have deemed root and size properties to be of importance. We have seen that the situation with respect to the root properties is more complex in some of the special cases and that we must take care when comparing polynomials with the same degree but different Galois group.

Chapter 6

Using subfield structure

We saw in chapter 5 that in some of the main special cases we have subfield structure in the algebraic number field used.

We are interested in methods which may speed up the number field sieve in the special cases for two reasons. Firstly, such a speed up would aid in the attempts to factor integers with known special structure such as those factored by the Cunningham project. In addition, in chapter 7 we will see that it is possible for some numbers with special structure to be isolated in an automated fashion without guidance as to the structure involved. That is, it is possible for a special case hitherto unrecognised as such by a human, and possibly used for cryptographic purposes, to be factored using a non-general set of parameters that result in a significantly reduced runtime from the general case.

We investigate the most obvious method of utilising the subfield structure available and then proceed to consider any implications for free relations, use of quadratic characters, the linear algebra and square root steps.

6.1 The Algorithm

Let n be a number with special structure and assume that we have produced two irreducible polynomials $f_1(X), f_2(X) \in \mathbb{Z}[X]$ by one of the methods seen to

produce number field sieve polynomials such that we have one or more subfields on the algebraic side. Let f_1 have either degree 4 or degree 6 (as these are the only composite degrees in the current range) and f_2 be a linear polynomial. In addition assume f_1 and f_2 have a common root $m \bmod n$. Let $\alpha \in \mathbb{C}$ be such that $f_1(\alpha) = 0$ and define a number field $K = \mathbb{Q}(\alpha)$. We also have a homomorphism $\varphi(\alpha) \equiv m \bmod n$.

In this chapter we will define $g_{(i,1)}$ to be the non-linear polynomial defining the ith subfield L_i of K and $g_{(i,2)}$ to be the corresponding linear polynomial.

Let K have a quadratic subfield L_1 defined by $(g_{(1,1)}, h_1)$. In the case that $d_1 = 6$ let there be in addition, a cubic subfield L_2 defined by $(g_{(2,1)}, h_2)$. We define $\hat{h}_i : \mathbb{Z}[\beta_i] \rightarrow \mathbb{Z}[\alpha]$ the subfield embedding, an injective homomorphism, induced by $\beta_i \mapsto h_i(\alpha)$, where $h_i(X)$ is the embedding polynomial. In a similar manner to standard NFS we construct a set S of integer pairs (a, b), a coprime to b and using each subfield construct sets S_i of integer pairs (a_i, b_i), a_i coprime to b_i for which

$$\prod_S (a - b\alpha) \prod_i \hat{h}_i\Big(\prod_{S_i}(a_i - b_i\beta_i)\Big) = \gamma^2 \in \mathbb{Z}[\alpha]$$
$$\prod_S (a - bm) \prod_i \Big(\prod_{S_i}(a_i - b_i h_i(m))\Big) = y^2 \in \mathbb{Z}.$$

We then have

$$\begin{aligned}
\varphi(\gamma)^2 &\equiv \varphi(\gamma^2) \\
&\equiv \varphi\Big(\prod_S (a - b\alpha) \prod_i \Big(\prod_{S_i}(a_i - b_i\hat{h}_i(\beta_i))\Big)\Big) \\
&\equiv \varphi\Big(\prod_S (a - b\alpha)\Big)\varphi\Big(\prod_i \Big(\prod_{S_i}(a_i - b_i h_i(\alpha))\Big)\Big) \\
&\equiv \prod_S (a - bm) \prod_i \Big(\prod_{S_i}(a_i - b_i\varphi(h_i(\alpha)))\Big) \\
&\equiv \prod_S (a - bm) \prod_i \Big(\prod_{S_i}(a_i - b_i h_i(m))\Big) \\
&\equiv y^2 \bmod n
\end{aligned}$$

so via the homomorphism we again have $\varphi(\gamma)^2 \equiv y^2 \bmod n$. We may then calculate $\gcd(n, \varphi(\gamma) - y)$ which will split n non-trivially in at least half the cases.

We will use the subfields with integral embeddings highlighted in the special cases mentioned in chapter 5. The embeddings preserve multiplicative and additive structure in the subfield hence ideals of the subfield map to ideals of the main field. In addition, first degree primes in the subfields either map under the subfield embedding to higher degree primes ideals in the main field or to ideals that factor in the main field.

However, in standard NFS we actively avoid working with $\gamma \in \mathbb{Z}[\alpha]$ divisible by higher degree primes. We do so as we require the factorisation of the norm, $\mathbf{N}(\gamma)$ to mirror the factorisation of γ. The obstruction when we allow higher degree primes in the standard case is that we may have

$$\prod_{p} p^e = \mathbf{N}(\gamma) = \mathfrak{N}\langle\gamma\rangle = \prod_{\mathfrak{p}} \mathfrak{N}(\mathfrak{p}^v)$$

with e even but v odd. This can happen, for instance, if a single $\mathfrak{p}$ in our product is a second degree prime.

In the subfield version of NFS we are effectively allowing higher degree primes in

$$\prod_{S}(a - b\alpha) \prod_{i} \hat{h}_i \left(\prod_{S_i}(a_i - b_i\beta_i) \right).$$

However, as we know that the primes from the subfield are higher degree primes in the main field (and to what degree) or factor in a known way in the main field we can easily sidestep this obstruction.

For instance: if we have a degree 4 main field and a degree 2 subfield we might have a relation in the subfield that contains a prime that maps to a second degree prime in the main field. It is treated as a single element that must itself appear to an even power in any set S_i that is produced. If a prime in a subfield factors in the main field then we effectively still treat that ideal as a single element requiring that all of the factors (and hence the ideal itself) appear to an even degree in the product.

We use the special form that was chosen in order that we did not encounter higher degree primes in standard NFS to retain a separation between those numbers which have only first degree primes as factors and those that may have higher degree primes. Thus we have opened up to us factor base elements which we

would not have been able to use in the past.

In order to see that this addition to standard NFS is coherent we need to reflect on some of the details of the algorithm. In the next section we run through the details of the implementation.

6.2 Implementation

The polynomial selection requires only that we select polynomials that permit subfield(s) on the algebraic side however this could have implications in some of the more technical steps of the algorithm. We examine each step in turn.

Sieving

The sieving takes place as in standard NFS in the main field and in a similar manner in the subfields. The only difference being that detailed above — that we need to use the subfield embedding to determine the auxiliary values to be tested for smoothness on the linear side in the subfield.

We construct the sets S, S_1 (and possibly S_2 if we have a second subfield) working in a similar way as in the main field. We will describe this for one subfield $L = \mathbb{Q}(\beta)$ which we will denote $(g_{(1,1)}, h_1)$. Let $g_{(1,2)}(X)$ be a linear polynomial with root $h_1(m) \bmod n$ and define the homogeneous polynomials $G_i(X, Y) = Y^{d_i} g_{(1,i)}(X/Y)$. We select the factor base $\mathcal{F}(B_i)$ of first degree prime ideals in an analogous manner to the main field. By sieving we find pairs (a, b) such that a, b are coprime and the integers $G_1(a, b)$, $G_2(a, b)$ factor over the primes in the subfield factor bases, except for any large primes, in the case of large prime relations, and call these subfield relations.

The introduction of large primes in the subfields does not pose any additional problems in terms of the collection of data; we work in the same way as in the main field. However, since we will have large prime relations from the main and subfield we may notice the effects of this in the filtering stage. We will discuss this shortly after introducing some of the technical apparatus that will allow us

to relate the main and subfield relations more freely.

Other than the method of obtaining the polynomials G_i with shared root modulo n we make the usual assumptions in the subfields and thus we are in exactly the situation described in chapter 2 and thus may sieve without impediment.

Free relations and the bridge

As in the main field we have free relations in the subfield. These relations occur when an integral prime p factors completely into first degree primes of the subfield.

If we have one linear polynomial and one of degree 2 (respectively 3) approximately 1/2! (respectively 1/3!) of the primes will give rise to free relations in the subfields. Hence a subfield of degree 2 is theoretically especially valuable. We would usually add these free relations in during the filtering stage as described in [13] if we wished to use them as some large factorisations still do for instance [18].

We note that when a prime p factors completely into first degree primes in more than one of the fields we effectively add two relations that allow interaction in the linear algebra stage between the previously distinct relation sets. We will refer to any relations that allow this as bridging relations.

We may go further than this. For each first degree prime ideal in a subfield that when mapped into the main field factors fully into first degree primes of the main field we have another form of bridging relation that carries extra information not previously expressed in the free relations. These relations are also effectively free to compute but they are immensely valuable since they are the only relations that describe the entire relationship between the main factor base and the subfield factor base — allowing a greater level of freedom in the linear algebra step. We will address the impact of these on the linear algebra later in this chapter.

In order to create these relations we need to be able to quickly and cheaply recognise when a prime from a subfield factors into first degree primes in the main field and identify the primes involved in the factorisation in the main field.

Let us restrict ourselves to monic polynomials (if the polynomials are not monic then the divisors of the leading terms, of which there is a finite and usually small number, must be treated specially but in other regards the theory below remains effectively the same). As we established earlier, in the field $K = \mathbb{Q}(\alpha)$ defined by f there is a one to one correspondence between pairs (p, r) and the first degree prime ideals, generated by p and $r - \alpha$. That is, we defined the factor base in the main field to be:

$$\mathcal{F}(B) = \{(p, r) \mid p \text{ prime}, p < B, r \in \mathcal{R}(p)\}$$

where

$$\mathcal{R}(p) = \{r \in \mathbb{Z}/p\mathbb{Z} \mid F(r, 1) \equiv 0 \bmod p\}.$$

Further we noted that $a - b\alpha$ is contained in $\langle p, r - \alpha \rangle$ iff $a - br \equiv 0 \bmod p$. We define the factor base in the subfields in a similar manner.

Under the embeddings into the main field, the first degree primes in the subfield may either map to higher degree primes or factor in the main field. Let us say that the field K has degree d and a subfield L of degree $d_L | d$ then the first degree primes in the subfield may either map to d/d_L-degree primes in the main field or factor into d/d_L first degree primes in the main field (there are other possibilities but we treat those as single entities that must themselves appear to an even degree in any resulting set S since they do not factor completely over the main field factor base).

Let us assume we have the field $K = \mathbb{Q}(\alpha)$ of degree 4 with a subfield $L = \mathbb{Q}(\beta)$ of degree 2 with embedding $\pm X^2$ as produced in chapter 5. The first degree primes in the subfield map to second degree primes in the main field or factor into two first degree primes in the main field.

For $p \in \mathbb{Z}$, p prime and less than the minimum of the factor base bounds, we have the following situations:

- If there are two first degree prime ideals corresponding to (p, r_i), $i = 1, 2$ in the subfield and none in the main field then on the embedding into the main field we will find that both of these are second degree primes and are generated by p and $r_i - \alpha^2$. No relations are added to the bridge.

- If there are two prime ideals of norm p in the subfield and two prime ideals of norm p in the main field then on the embedding into the main field we will find that one prime of the subfield maps to a second degree prime in the main field and the other factors into two first degree primes.

  We recognise these in the following manner: if the prime ideal $\langle p, r - \beta \rangle$ maps under the embedding X^2 to the ideal $\langle p, r - \alpha^2 \rangle = \langle p, r_1 - \alpha \rangle \langle p, r_2 - \alpha \rangle$ where $-(r_1 r_2) \bmod p = r \bmod p$ then the prime corresponding to (p, r) in the subfield factors into the primes corresponding to (p, r_1) and (p, r_2) in the main field. Since we already have the primes below some bound B for the main field we are able to quickly identify which primes are involved. We add one relation to the bridge at the cost of one modular multiplication and some checking.

- If there are two primes in the subfield and four primes in the main field then on the embedding into the main field we will find that both subfield primes factor into two first degree primes of the main field. Hence we add two relations to the bridge. These relations are added instead of the two free relations we would usually add in this case so we do not increase the size of the bridge but we do add two new pieces of information (both individual factorisations as well as the full factorisation).

  We use the same method as above to find the factors. This will involve at most three modular multiplications and some checking.

- A finite number of primes (those that divide the discriminant) will act slightly differently and should be considered on a case by case basis.

In the case of a degree 6 field with two subfields, one of degree 2 with embedding $\pm X^3$ and one of degree 3 with embedding X^2, let a triple (a, b, c) describe a situation in which we have a primes in the degree 6 main field, b primes in the degree 3 subfield and c primes in the degree 2 subfield.

For $p \in \mathbb{Z}$, p prime and below the factor base bounds we can have the following situations:

- In the cases $(0, 0, 2)$, $(0, 1, 0)$ and $(0, 3, 0)$ the prime ideals of the subfields do not factor in the main field. In the first case the first degree primes of the degree two subfield map to degree three primes in the main field. In

the latter two cases the first degree primes in the subfield map to second degree primes in the main field. No relations are added to the bridge.

- In the case $(2, 1, 2)$ the first degree prime in the degree 3 field factors into the two first degree primes in the main field. The first degree primes in the degree 2 subfield (which are of degree 3 in the main field) do not factor completely into first degree primes in the main field (though each one has a divisor, which we may identify if required, in the main field but we cannot use that information in the bridge).

  Hence we add one relation to the bridge. This requires no computation.

- In the case $(6, 3, 2)$ the first degree primes in the degree 2 subfield factor into three first degree primes in the main field in a way analogous to the $d = 4$ case.

  Each of the three first degree primes in the degree 3 subfield split into two first degree primes in the main field.

  These five relations taken together represent the information that would usually have been added in the free relations corresponding to these primes in each field (of which there are three). Hence we have added two extra relations but a finer grade of information regarding the structure of the ring is apparent.

- We may have different (a, b, c) in some cases where the prime divides the discriminant of f and we work on a case by case basis.

Quadratic characters

We must show that quadratic characters may still be used to good effect to ensure that we are able to produce a square of an element in the algebraic number field K.

As we saw earlier quadratic characters are used to ensure that we have a square of an element of $K = \mathbb{Q}(\alpha)$ with high probability. We still need to use such a facility. Quadratic characters are defined on all elements of $\mathbb{Z}[\alpha]$ and so we need not extend the character function.

We work as in standard NFS with the proviso that we select prime ideals that are in none of the factor bases and that do not divide any factor base element (note that some of the elements in the subfields factor when embedded in the main field so we must take extra care here). We work with ideals of the main field since we aim to find a square in $\mathbb{Q}(\alpha)$.

We generalise proposition 2 to encompass the case in which we are working.

Proposition 4 *Assume that we have sets S and S_i as above. S is from the main field and so has only first degree primes as divisors. S_i are formed of relations that come from the subfields and so, under the embedding into the main field, may have higher degree prime divisors. Due to the presence of the bridging relations we cannot assume that the individual products:*

$$\prod_S (a - b\alpha), \quad \prod_{S_i} (a_i - b_i h_i(\alpha)),$$

are themselves squares. We assume only that

$$f'(\alpha) \prod_S (a - b\alpha) \prod_i \prod_{S_i} (a_i - b_i h_i(\alpha))$$

is the square of an element of $K = \mathbb{Q}(\alpha)$.

We select a set of odd prime numbers p such that:

$$\begin{aligned} a - br &\not\equiv 0 \bmod p, && \forall\ (a,b) \in S, \\ \forall i \quad a_i - b_i h_i(r) &\not\equiv 0 \bmod p, && \forall\ (a_i, b_i) \in S_i, \end{aligned}$$

we also assume that $f'(r) \not\equiv 0 \bmod p$.

Then we have

$$\prod_S \left(\frac{a - br}{p}\right) \prod_i \prod_{S_i} \left(\frac{a_i - b_i h_i(r)}{p}\right) = 1$$

Proof:

Let $\mathbb{Z}[\alpha] \to \mathbb{Z}/p\mathbb{Z}$ be the ring homomorphism mapping $\alpha \mapsto r \bmod p$ and let $\mathfrak{p}$ be the first degree prime corresponding to (p, r).

Define the map $\chi_p : (\mathbb{Z}[\alpha] - \mathfrak{p}) \to \{\pm 1\}$ to be the composition of

1. $(\mathbb{Z}[\alpha] - \mathfrak{p}) \mapsto (\mathbb{Z}/p\mathbb{Z} - 0)$ with
2. The Legendre symbol $(\mathbb{Z}/p\mathbb{Z} - 0) \to \{\pm 1\}$.

Clearly we have

$$\chi_p(a - b\alpha) = \left(\frac{a - br}{p}\right) \qquad \chi_p(a_i - b_i h_i(\alpha)) = \left(\frac{a_i - b_i h_i(r)}{p}\right)$$

As we saw earlier we have

$$f'(\alpha) \prod_S (a - b\alpha) \prod_i \prod_{S_i} (a_i - b_i h_i(\alpha)) = \delta^2$$

for some $\delta \in \mathbb{Z}[\alpha]$. The factors on the left are not in $\mathfrak{p}$ since if we have, for instance, $a_1 - b_1\beta \in \mathbb{Z}[\beta]$ this maps under the subfield embedding $\hat{h}_1 : \mathbb{Z}[\beta] \to \mathbb{Z}[\alpha]$ to $a_1 - b_1 h_1(\alpha) \in \mathbb{Z}[\alpha]$. Elements of $\mathbb{Z}[\alpha]$, $\sum_i a_i \alpha^i$, are members of $\mathfrak{p}$ if and only if $\sum_i a_i r^i \equiv 0 \bmod p$ but we have assumed that $a_1 - b_1 h(\alpha) \not\equiv 0 \bmod p$ and $a_2 - b_2 h_2(\alpha) \not\equiv 0 \bmod p$. Hence we have $\delta \notin \mathfrak{p}$. Applying χ_p to the equation finishes the argument □

As before we are actually interested in the converse of this result. From [12] we have that if $\delta \in \mathbb{Z}[\alpha] - 0$ satisfies $\chi_p(\delta) = \left(\frac{\delta}{p}\right) = 1$ for all first degree primes $\mathfrak{p}$ with $2\delta \notin \mathfrak{p}$ or even for all such $\mathfrak{p}$ with finitely many exceptions then δ is a square in K. No assumption is made about the form of δ hence we may use quadratic characters to achieve the same results except that we must take more care when selecting the prime ideals.

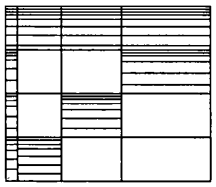
Filtering and linear algebra

We must assess any effect on the filtering and linear algebra stages including the use and effect of any free relations.

Once we have collected relations in the main field and any subfields we filter the data and then we build a matrix. The main and subfields each have an algebraic factor base and a rational factor base of primes in $\mathbb{Z}$, the matrix contains a row

for each factor base element in any of the two (three) algebraic factor bases and for each rational prime that is in any of the two (three) rational factor bases.

Each column corresponds to a relation as before. A relation from the main field will only involve primes in the main algebraic factor base and the rational factor base and similar for the subfield relations.



Thus the matrix has blocks in which there are no non-zero entries. The bridging relations are an exception, each bridging relation will contain an entry for one rational prime and entries corresponding to a mixture of main and subfield primes.

There are a few comments we must make regarding the effects of this matrix structure in the filtering and linear algebra stages.

We remove duplicate relations in the usual fashion. In the case of singleton removal we must note that first degree primes of the main field can divide primes in the subfields. We must take care not to remove relations that contain a prime present only once in the subfield but whose factors are present multiple times in the main field. This is accomplished by adding in the bridging relations prior to this step.

In the merging step of the algorithm the bridge relations have very low density and so should be treated as such in a similar manner to the treatment of the free relations in the standard filtering case. We may choose not to allow merges of relations from a subfield and the main field as this will cause fill in in the blocks of zeros. More importantly we do not know what other effects such merging would have on the density of the matrix or the complexity of the algorithm.

If the subfield method is shown to be viable for realistic sized factorisations then the details of the filtering will become more important and should be investigated.

After the filtering comes the linear algebra stage. If no merging has occurred the matrix will have blocks of zeros as mentioned above, this may be beneficial in the linear algebra stage as it is possible that the density of the matrix will be lower than is usually the case or that we can take advantage of the structure in some other manner. If there has been merging then there will have been a certain amount of infill in these blocks. In the linear algebra step we proceed as usual.

The only important point to note is that this allows a prime $\mathfrak{q}$ in the subfield which factors as $\mathfrak{p}_1\mathfrak{p}_2$ in the main field to either be found in our output square ideal to an even power $\mathfrak{q}^2$ as a single (subfield) entity in its own right or we to be found as $\mathfrak{q}^2\mathfrak{p}_1^2\mathfrak{p}_2^2$ by involving the bridging relation that contains $\mathfrak{q}$, $\mathfrak{p}_1$, $\mathfrak{p}_2$ (and the corresponding p to the appropriate power on the linear side). This allows a prime that occurs only once in a set of subfield relations but factors entirely into first degree primes in the main field to be involved in the final product. This method will not allow the situation in which a higher degree prime of the main field, say degree 2, appears alone in the product. Such a factor base element will not appear in any of the bridging relations and hence is treated as a single entity that must itself appear as a square if it is to be of any use.

Square roots

Finally we will need to be able to calculate square roots of algebraic numbers of a more general form.

The final step in the number field sieve is the square root on the algebraic side. We now have to square root algebraic elements of the form

$$\prod_S (a - b\alpha) \prod_i \hat{h}_i\Big(\prod_{S_i}(a_i - b_i\beta_i)\Big)$$

This is not a barrier since the method due to Montgomery [39, 73] does not specify a particular form for the algebraic element.

Montgomery's technical report [73] gives this algorithm in a more general form — there is no assumption that we have a product of elements of the form $a - b\alpha$ or that only first degree primes are involved. Montgomery assumes that

1. $\gamma = \prod_i g_i(\alpha)$ is a product which we think is a non-zero square in $\mathbb{Q}(\alpha)$.
2. We have the prime ideal factorisation of γ and, in fact, of each $\langle g_i(\alpha)\rangle$.
3. Each prime ideal has an even exponent in the factorisation of $\langle\gamma\rangle$.

In this situation the algorithm provided by Montgomery will allow us to construct

the square root, if it exists, using the ideal factorisations.

As detailed above we are able to use the embeddings to provide us with a prime ideal factorisation in the main field of each of the $\langle g_i(\alpha)\rangle$, hence we may use Montgomery's method without impediment.

6.3 Theoretical expectations

The size of the numbers we wish to be smooth over the factor base in the main field is $|F_1(a,b)F_2(a,b)|$ while in a subfield we are interested in $|G_1(a,b)G_2(a,b)|$. If we consider the particularly common situation where $f_1(X) = cX^d - t$, $f_2(X) = X - m$, d composite with a factor d_s such that there is a subfield defined by $(cX^{d_s} - t, h(X))$. Hence in the main field the numbers that we wish to be smooth are

$$|(a - bm)(ca^d - b^d t)| \leq (a_{\max} + b_{\max} m)(|c|a_{\max}^d + b_{\max}^d |t|)$$

if we assume $a_{\max} = -a_{\min}$. In the subfield we have

$$|(a - bh(m))(ca^{d_s} - b^{d_s} t)| \leq (a_{\max} + b_{\max}|h(m)|)(|c|a_{\max}^{d_s} + b_{\max}^{d_s}|t|).$$

On the algebraic side the size of numbers $|c|a_{\max}^{d_s} + b_{\max}^{d_s}|t|$ is favourable when compared to the main field and it would suggest that the subfield provides beneficial structure. However, on the linear side we have $a_{\max} + b_{\max}|h(m)|$ where h is the embedding polynomial which is at least as large as $h(X) = X^2$. We immediately see the problem: the numbers we wish to be smooth over the rational factor base will be far too large to have expectations that this algorithm will perform well.

It seems unlikely that we would be able to compensate for this by increasing the size of the rational factor base and decreasing the sizes of the algebraic parts. We investigate this in the next section.

On the other hand the algebraic side is smaller and hence more likely to be smooth. We see further support in the following section that if a method of utilising the structure were found, this could have a significant effect in practice.

We may also consider that the matrix produced may be sparser than usual and of a differing form. It is possible that this could be utilised during the practical bottle neck — but this is unlikely to be of use if it is necessary to unduly prolong the sieving step in order to aid us in the linear algebra step.

We leave for further research the question of whether there is any way to make use of this advantage on the algebraic side without triggering a similar explosion in the coefficient on the linear side. Another possibility may be that of working with two non-linear polynomials. However there is no immediate method for choosing a pair of non-linear polynomials with a shared root modulo n such that any subfield structure present can be utilised. We might also consider working in field extensions, rather than subfields, of the main field.

6.4 Practical results

We provide three sorts of practical support for the theory above. Firstly we complete an example factorisation to illustrate the issues encountered with the method. Secondly we consider a number of more realistic size and note the failure of the method in line with the theory. We then provide additional support for this by making use of the techniques established in chapter 4 for improved estimation of the quantity of raw data we might expect to collect.

6.4.1 Some example factorisations

For illustrative purposes we include the following example factorisation, the number factored is small compared to those usually factored using the number field sieve but should serve as a starting point for analysis of the method. In fact we will see immediate support for the fact that this method of using the subfield structure is unlikely to be of practical use. The number we will illustrate the method with is $n = 2^{149} - 1$. Since we wish to consider the subfield method we need to use composite degree. In this case we will use degree 4, though the asymptotics might suggest degree 3 (or in practice MPQS) for this number. We should bear this in mind since it could mean that the subfield becomes more useful than it would otherwise have been. However we will see that even in this case,

	main field
$f_1(X)$	$X^4 - 8$
$f_2(X)$	$X - 2^{38}$
Region	$\|a\| \leq 10^4$
	$1 \leq b \leq 10^4$
B_1	10^4
B_2	10^4
Relations	3417

Figure 6.1: Standard NFS

	main field	subfield
$f_1(X)$	$X^4 - 8$	$X^2 - 8$
$f_2(X)$	$X - 2^{38}$	$X - 2^{74}$
Region	$\|a\| \leq 10^4$	$\|a\| \leq 10^4$
	$1 \leq b \leq 10^4$	$1 \leq b \leq 10^4$
B_1	6362	6272
B_2	6204	6204
Relations	1293	409

Figure 6.2: Subfield NFS $(1/3, 1/3, 1/3)$

	main field	subfield
$f_1(X)$	$X^4 - 8$	$X^2 - 8$
$f_2(X)$	$X - 2^{38}$	$X - 2^{74}$
Region	$\|a\| \leq 10^4$	$\|a\| \leq 10^4$
	$1 \leq b \leq 10^4$	$1 \leq b \leq 10^4$
B_1	4520	4568
B_2	9804	9804
Relations	1403	340

Figure 6.3: Subfield NFS $(1/2, 1/4, 1/4)$

where we might reasonably expect the subfield to be of some use, the method fails to be practical.

Since our objective is merely an illustration of the differences in performance of the main and subfield sieving over the same sieve region as we alter the factor base bounds we have used fairly artificial parameters and no large primes. We have not included the extra bridging relations but have included the usual free relations. We use a more realistic set of parameters and large primes in the following experiment.

We compare standard NFS with two different parametrisations of the subfield method. We keep the total number of factor base elements approximately equal in all three cases. In the first subfield example we allow 1/3 of the factor base elements in the rational factor base, 1/3 in main field and 1/3 in the subfield. In the second case we allow 1/2 of the factor base elements in the rational factor base and 1/4 each in the others. As we can see the quantity of relations found in the subfield is extremely small. This leads to us not collecting enough relations when using subfields.

In fact the situation is worse than it looks since the majority of the relations

found in the subfields are in fact free relations (the small size of the Galois group means that we have more of these than in the main field: this seems beneficial in general but in fact we may only take advantage of these if we have enough non-free relations in the subfield).

6.4.2 Sieving tests

Other trial factorisations of n of the size used above produce similar results. However, we would like to consider whether allowing large primes and a less unrealistic parametrisation leads to any substantial difference in our conclusions. Ideally we would attempt a degree 6 factorisation. Unfortunately, lacking the resources to complete any substantial sieving in a realistic time for a degree 6 field with parameters of an appropriate size we instead consider another degree 4 field. As we shall see, the results are of a definite nature and it would seem that sieve trials in a degree 6 case would not have added any further insights.

The degree 4 field is one that has previously been selected and used to factor a cofactor (of 106 digits) of $2^{543} - 1$ [39]. The defining polynomials are $f_1(X) = 4X^4 + 2X^2 + 1$, $f_2(X) = X - 2^{90}$. The degree 4 main field has one subfield $(4X^2 + 2X + 1, X^2)$. We sieved the lines $b = 1$ to $b = 10$ with an interval $|a| \leq 3.5 \cdot 10^5$, prime bounds of $B_1 = B_2 = 5 \cdot 10^5$ and large prime bounds of $L_1 = L_2 = 12 \cdot 10^6$. We used identical parameters in both the main field and subfield. In the main field a total of 130 relations were collected while in the subfield we found no relations whatsoever. The small size of the polynomials and the small b values mean that this should have been the most fertile part of the sieve region in this case. The size of the numbers on the linear side of the subfield were simply too large. Hence we ceased any further experimentation regarding altering the parametrisations as it became clear that the impact of the size of the linear polynomial corresponding to the subfield was not likely to be overcome by any moderate change in the prime bounds.

In contrast, sieving just the algebraic sides in the main field and subfield (with identical parameters to those above) produced 72499 relations in the main field and 1308476 in the subfield. This amply illustrates the possibilities on the algebraic side which originally inspired us to investigate this method. Unless the subfield method can be used in a situation where the effect of the explosion in size

on the subfield linear side can be suppressed (or the use of a linear side avoided altogether) it does not appear that this method can be practical.

6.4.3 Estimating yield

The above method of working is favoured by selecting for a given n a value of d for the main field which is greater than that we would usually select since the subfield parameters will then be closer to those that the asymptotics suggest for use with n. We are unable to carry out any significant quantity of sieving for degree 4 or 6 factorisations due to the resources that are required. To provide additional support for the above results we will also appeal to estimates of the quantity of data produced based on the methods discussed in chapter 4.

We examine some factorisations which have been completed with $d = 6$ and estimate the quantity of relations which would be found in the subfields had they been sieved over the same region. As the estimation method employed (that of chapter 4) may not be used with lattice sieve parameters we work only with the classical or line sieved component.

We work with the factorisations detailed in figure 6.4, for the first we have sieving data for which the linear and non-linear sides were sieved separately over a small part of the sieve region. For the latter two the estimates reflect the entire region.

n	Gal(f)	Original polynomials	Subfields	Source
$2,773+$	$D(6)$	$X - 2^{129}$	(X^2+2, X^3)	[14, 15, 20]
SNFS-233		$X^6 + 2$	(X^3+2, X^2)	
$3993M$	$F_{18}(6)$	$3^{55}X - 1$	(X^2+3X+3, X^3)	[14, 20]
		$X^6 + 3X^3 + 3$	—	
$10,211-$	$D(6)$	$X - 10^{35}$	$(10X^2-1, X^3)$	[16]
SNFS-211		$10X^6 - 1$	$(10X^3-1, X^2)$	

Figure 6.4: Details of factorisations used with estimation tests

In the estimates below all the linear polynomials were considered to have $\alpha(F_2) = 0.569915$. We allowed up to 2 large primes on each side (in the first case we discounted the 3-partials and in the final factorisation it is not clear how many large primes were permitted during the sieving as they are not reported separately).

Looking at figure 6.5, firstly we note the rather poor estimate on the non-linear

name	2,773+ linear	2,773+ non-linear	$3993M$	SNFS-211
A	28875000	28875000	$168 \cdot 10^4$	$6 \cdot 10^6$
B	[1000001, 1000100]	[1000001, 1000100]	$156 \cdot 10^4$	$18 \cdot 10^6$
B_1	$2 \cdot 10^7$	—	$44 \cdot 10^5$	2^{24}
B_2	—	$2 \cdot 10^7$	$11 \cdot 10^6$	2^{24}
L_1	10^9	10^9	$6 \cdot 10^7$	$6 \cdot 10^8$
L_2	10^9	10^9	$6 \cdot 10^7$	$5 \cdot 10^8$
$\alpha(F_2)$	—	1.938592	1.468072	1.331229
$\alpha(G_1)$	—	1.119345	1.504661	1.190318
$\alpha(G_2)$	—	1.572525	—	1.306183
		Total relations:		
Main field:				
Sieved:	634590	1657934	5975620	23510939
Estimate:	624150	1439679	5969603	20863647
Subfield 1:				
Estimate:	0	3475671224	0	0
Subfield 2:				
Estimate:	0	488547479	—	2494

Figure 6.5: Estimates of yield in the main and subfields

side of 2,773+ at only 86% of the actual reported relations. This is almost certainly a side effect of the very small b interval which allowed us to use a maximum of $K = 8$ (64 subregions) in the estimate, it seems likely that this is another case where we would benefit from splitting the a interval and b interval using different K values. In SNFS-211, we do not have enough information regarding the sieving completed to assess why our estimate is 89% of the total which, while being a reasonable estimate, could clearly be improved. However, despite this the results in the table show overwhelming support for the conclusion that the subfield method as described above, while a coherent method that would appear to take advantage of subfield structure is not practical and that the key reason for this is the size of the auxiliary numbers on the linear side in the subfield.

6.5 Summary

We considered the most natural extension of the number field sieve to fields with subfields in an attempt to utilise the structure to our advantage. We gave an overview of the changes to the algorithm in this case focusing on showing that this was coherent.

Theoretical and experimental evidence suggests that the subfield method discussed is not practical. We leave to further research the question as to whether the clear advantage noted on the algebraic side alone can be successfully utilised in some other fashion to provide a practical advantage.

Since the core method is not practical we do not elaborate on issues such as large primes or filtering leaving such concerns to be dealt with if a variant of the method should prove more useful.

Chapter 7

Polynomial selection: special versus general

In this chapter we are interested in whether any form of automatic polynomial selection can expose a special case. From this point onwards we will cease to define special cases by the method in which they were produced but instead will define a "special" case of the number field sieve as one which has the pleasant characteristics discussed in the chapter 5. That is, any polynomial which is abnormally small with respect to the size of n, or abnormally small in the higher coefficients (accompanied by a skewed region over which to work) and with Galois group which is strictly smaller than the full symmetric group. Notice that this encompasses but extends the original meaning of the word "special" and all of the special number field sieve factorisations in the literature where the polynomial used or a method of obtaining polynomials is reported (see chapter 5 and appendix A). We are particularly interested in the small size of the Galois group when the degree chosen is composite, in this situation we may also uncover subfield structure.

However, if we randomly select polynomials we will almost always have Galois group equal to the full symmetric group. It has been assumed in the past that methods used to select GNFS polynomials such as the much lauded methods described by Murphy [76] produce "random" polynomials and that we may further assume that the Galois group will be the symmetric group S_d [76, chapter 3].

Can Murphy's methods isolate special case polynomials? Or should integers be tested in another manner prior to using Murphy's methods to produce a polynomial? If methods such as Murphy can do as well or better than a human with no extra information on any structure in the integer to be factored then this throws up questions regarding the asymptotics — after all, past special case factorisations are significantly faster based on a perceived reliance on extra information provided by a human. If structure is present but we need not rely on a human to provide this information and hence reduce the time to factor the number then this brings about questions regarding the accidental or deliberate use of numbers with special structure, perhaps not easily noted by a human, in RSA.

7.1 Some open questions

A series of results and conjectures summarised by Malle [66] suggest that while monic polynomials of degree d with Galois group different from S_d are rare that it is not rare, in some sense, for a number field of degree d to have Galois group $G \neq S_d$. In particular Malle notes the following proposition:

Proposition 5 (Malle) *Let K be a number field and $d = 2k > 2$ or $d = 3k > 3$. Then the number of field extensions of K of degree d with Galois group not the symmetric group grows at least linearly with the absolute value of the discriminant. In particular, there exists a transitive subgroup $G < S_d$ with $Z(K, G; x) \geq cx$ for an unbounded set of values of x.*

where

$$Z(K, G; x) := \left|\{L/K \mid \mathrm{Gal}(L/K) = G, \mid \mathcal{N}_{K/\mathbb{Q}}(\mathrm{Disc}(L/K)) \mid \leq x\}\right|.$$

Here Malle uses $\mathrm{Gal}(L/K) = G$ where L/K is a field extension such that the Galois group of the Galois closure $\hat{L}/K$, viewed as the permutation group on the set of embeddings of L into $\hat{L}$, is permutation isomorphic to G. It is known that the number of extensions of K with norm of the discriminant bounded by x is finite hence $Z(K, G; x)$ is finite. Further to this Cohen notes the following proposition [22, Proposition 9.3.4]:

Proposition 6 *The number of non-isomorphic number fields of fixed degree d and discriminant in absolute value bounded by x is at most equal to $cx^{(d+2)/4}$ for some constant c depending only on d.*

It is known that this bound is not the best possible for $d = 3$. Cohen notes that it seems unlikely to be the best possible for $d = 4$ and reports a conjecture which would sharpen the bound to cx as $x \longrightarrow \infty$ for some constant c dependent on d.

In particular it is known that number fields with Galois group D_4 satisfy

$$Z(\mathbb{Q}, D_4; x) \sim c(D_4)x$$

where $c(D_4) \approx 0.0523$ [23]. If the conjecture holds then number fields with $G = D_4$ would have positive density.

If $d = 4$ or $d = 6$ is taken as the degree of our number field then the above might be of peculiar interest should we produce a method by which we can make advantageous use of any subfield structure present. This leads us to note two open questions:

Question 1 *Are polynomials produced by Murphy's schema "random" i.e. do the same (as for all polynomials) asymptotics apply?*

Question 2 *To what extent is Murphy's schema selecting fields rather than polynomials?*

7.2 Polynomial selection in the general case

In chapter 3 we described Murphy's methods for creating a list of polynomial pairs with better than average properties. In addition we recounted his method for selecting better polynomial pairs from such a list (without utilising expensive sieve tests).

We wish to choose m and f_m or some variant which preserves $f_m(m) \equiv 0 \bmod n$ with good combinations of size and root properties and we are interested in what

type of polynomials and number fields these methods can produce. We will require some definitions.

Recall the original base-m general polynomial selection method from chapter 2; define the polynomial produced in this manner to be the (primary) base-m representation of n, $n = \sum_{i=0}^{d} a_i^{(m)} m^i$, $0 \leq a_i^{(m)} < m$.

- Many authors note that heuristically it appears to be a good idea to adjust the $a_i^{(m)}$ to lie between $-m/2$ and $m/2$ (this can reduce the maximum size of the coefficients). If the $a_i^{(m)} > \lfloor m/2 \rfloor$ then set $a_i^{(m)}$ to be $a_i^{(m)} - m$ and set $a_{i+1}^{(m)}$ to be $a_{i+1}^{(m)} + 1$. Let $f(X) = \sum_{i=0}^{d} a_i X^i$ be the polynomial whose coefficients are the $a_i^{(m)}$ reduced in this way working from $i = 0, \ldots, d$ through the coefficients.

  We refer to this polynomial as the adjusted base-m representation.

- Murphy defines translations and rotations of (adjusted) base-m polynomials:

  - Translation by t: $f_t(X) = f(X - t), f_t(m_t) \equiv 0 \bmod n$, $m_t = m + t$.
  - Rotation by $R(X)$: $f_R(X) = f(X) + R(X)(X - m)$ (same m).

  Murphy uses linear R, Gower uses higher degree rotations, with the degree of R less than the degree of f.

- We define a base-m variant to be any polynomial produced from either a base-m representation or an adjusted base-m representation by any sequence of translations and rotations. We note that a base-m variant will have the same degree as the primary base-m representation except in the case that an adjustment process ends with the coefficient $a_d^{(m)} > \lfloor m/2 \rfloor$ which is a rare occurrence.

- A primary or adjusted (resp. variant) base-m polynomial will be called χ-small when χ is the largest value of $|a_i|/m$ (resp. $|a_i|/m_t$ where m_t is the appropriate translation of m) for $i = 1, \ldots, d$. If the value of χ is unimportant we refer to the polynomial as small.

We are interested in whether the special case polynomial pairs selected by humans (with knowledge of special structure) could be produced as output from the general case algorithm.

In order for a polynomial pair to be selected by Murphy's skewed or non-skewed schema it is necessary for the non-linear polynomial to be a base-m variant. In addition it must be relatively small: in the case of the non-skewed method, χ-small for χ within the chosen bounds; in the case of the skewed method the polynomial should have small higher coefficients. Finally, the leading coefficient should be divisible by powers of many small primes. However, we may see a compromise in the case of root properties if this favours extremely small size (it is less likely that size will be compromised for root properties which tend to have a less marked overall effect but it is possible). In fact, the most obvious characteristic of special case polynomials is that they are extremely small and it is more unusual for such polynomials to have a leading coefficient that is of the required type.

Typical special case polynomials f with root $m \bmod n$ produced by hand using knowledge of special structure are not naturally base-m representations for the particular m.

For instance, let us take a special case $n = r^e - s$ with polynomial f formed using the original SNFS method. f is of the form $f(X) = X^d - t$, with $m = r^k$ such that $m^d - t \equiv 0 \bmod n$ and hence $m^d - t = ln$ some $l \in \mathbb{N}$, this situation is a common occurrence amongst the special forms. The polynomial f is not a base-m representation unless $l = 1$ ($\Leftrightarrow d|e$ and $t = s$). If $l \neq 1$ then since t is small with respect to n (chapter 5) we have $m^d > n$ (perhaps significantly so) and so we cannot form a base-m representation of degree d. Even if $l = 1$ we would still require that $t < 0$ (and hence $s < 0$) otherwise we have that $m^d > m^d - |t|$ and hence that the standard base-m representation would in fact produce a degree $d - 1$ polynomial.

For other special f we have similar arguments — although in some cases f is a base-m representation this was never the intention and is often not the case. In such cases f also cannot be produced as a rotation or adjustment of a primary base-m representation since both preserve the value of m and the degree — we would have to have started with a base-m representation of degree d but we have already seen that no such thing can exist. This will hamper the production of such polynomials by any method that starts by producing a primary base-m representation and then manipulating it.

Translations alter not only the polynomial but also the m value and so it is possible that such polynomials could occur as a translation. Translations are used alongside rotations in an attempt to skew the polynomial and reduce its size over the sieve region (while keeping the resulting polynomial central on the X-axis).

In cases where the special structure polynomial f is a (adjusted) base-m polynomial we are still unlikely to produce f as output unless the leading coefficient is of the desired type. However there are natural non-monic variants of these SNFS polynomials that do still betray the special structure inherent in the number. Instead of attempting to produce the precise pair that a human would select we instead examine variants which can be produced. We consider whether for numbers with some special structure, Murphy's schema can produce, without access to that structure, polynomials that have some or all of the special characteristics noted in chapter 5, that is, extremely small polynomials with Galois group strictly smaller than the full symmetric group.

7.3 Special case variants

Let $n = r^e - s$ and define

$$\begin{aligned} k &= \left\lceil \frac{e}{d} \right\rceil, \quad m = r^k, \quad f(X) = X^d - sr^{kd-e} \\ k_1 &= \left\lfloor \frac{e}{d} \right\rfloor, \quad m_1 = r^{k_1}, \quad f_1(X) = r^{e-k_1 d} X^d - s \end{aligned}$$

Note that $k = k_1 + 1$ and that

$$f_1(m_1) = r^{e-k_1 d} r^{k_1 d} - s = r^e - s = n = 0 \bmod n.$$

Fix roots of f and f_1: $f(\alpha) = 0$, $f_1(\beta) = 0$ then $K = \mathbb{Q}(\alpha)$, $K_1 = \mathbb{Q}(\beta)$ are isomorphic number fields (and hence have the same Galois group): For K and K_1 to be isomorphic we must be able to write some root, say α of f as $P(\beta) \in \mathbb{Q}[X]$ with P of degree less than d. In other words we require that both $\alpha, 1, \beta, \beta^2, \ldots, \beta^{d-1}$ and $\beta, 1, \alpha, \alpha^2, \ldots, \alpha^{d-1}$ are linearly dependent (for particular roots α of f and β of f_1). Now $\alpha^d = sr^{kd-e}$ so α is one of the dth roots of sr^{kd-e},

$\beta^d = s/r^{e-k_1 d}$ and hence β is one of the d^{th} roots of $s/r^{e-k_1 d}$. We see that α and β are themselves linearly dependent by examining $\alpha - r\beta$:

$$\begin{aligned}
\alpha - r\beta &= s^{1/d} r^{(kd-e)/d} - r s^{1/d} r^{-(e-k_1 d)/d} \\
&= s^{1/d} r^{(kd-e)/d} - s^{1/d} r^{(d-e+k_1 d)/d} \quad \text{and } k = k_1 + 1 \\
&= s^{1/d} r^{(kd-e)/d} - s^{1/d} r^{(d-e+kd-d)/d} \\
&= s^{1/d} r^{(kd-e)/d} - s^{1/d} r^{(kd-e)/d} \\
&= 0 \quad \text{as required.}
\end{aligned}$$

In fact we can generalise this by taking $k = k_c + c$ whereupon we have the polynomial $f_c(X) = r^{e-k_c d} X^d - s$, $m_c = r^{k_c}$ such that

$$f_c(m_c) = f(r^{k_c}) = r^{e-k_c d} r^{k_c d} - s = n = 0 \mod n$$

again this defines an isomorphic number field:

$$\begin{aligned}
\alpha - r^c\beta &= s^{1/d} r^{(kd-e)/d} r^c s^{1/d} r^{-(e-k_c d)/d} \\
&= s^{1/d} r^{(kd-e)/d} - s^{1/d} r^{(cd-e+k_c d)/d} \\
&= s^{1/d} r^{(kd-e)/d} - s^{1/d} r^{(cd-e+kd-cd)/d} \\
&= s^{1/d} r^{(kd-e)/d} - s^{1/d} r^{(kd-e)/d} \\
&= 0 \quad \text{as required.}
\end{aligned}$$

Since in forming these polynomials we take $1 < k_c < k$ and, more importantly have that $f_c(m_c) = n$ we have $m^d = r^{k_c d} < r^e - s$ (for some k_c at least, providing s is not excessively large, — but also see later). Hence these are often base-m expansions. We will refer to them as the c^{th} variant of the SNFS original. The factor base produced by the c^{th} SNFS-variant will be almost identical in terms of the distribution of primes to that produced by the original SNFS polynomial. The only difference occurs at primes which divide r.

In addition we may form translations of these variants. Let $f_{\text{var}}(X) = r^{e-k_{\text{var}} d} X^d - s$ with $f(m_{\text{var}}) = 0 \mod n$ be defined as above, then we can define translations

$$f_t(X) := f_{\text{var}}(X - t) = r^{e-k_{\text{var}} d}(X - t)^d - s, \quad t \in \mathbb{Z}$$

which will have all of the coefficients except the constant term divisible by $r^{e-k_{\text{var}} d}$.

Such polynomials will produce isomorphic number fields with the same Galois group. We can see this immediately: if $f_{\text{var}}(\alpha) = 0$ and $f_t(\beta) = 0$ then $\beta = \alpha - t$ hence α, β and 1 are linearly dependent and the number fields will be isomorphic.

We are also interested in how large the value of s could grow while still retaining this structure (that is, retaining the Galois group and the small, special structure coefficients except in the constant term). Consider $n = r^e + s$, $s > 0$ and assume that we have some base-m polynomial $f(X) = \sum_{i=0}^{d} a_i X^i$, such that $f(m) \equiv 0 \mod n$, for this number. In order to find a base-m polynomial for the integer $n + 1$ we must only add 1 to the constant term of this polynomial. We may continue this way until the constant term reaches the size m at which point we increment the coefficient a_1 by 1 and zero the constant term, we then continue to increment the constant term. These are all clearly base-m polynomials however they will not all satisfy our additional criteria. Putting size matters aside for one moment we are interested to see at what point the Galois group changes. It is the structure in the higher coefficients that induces the smaller Galois group, any number that we add that alters any coefficient but the constant term will result in the Galois group S_d (with probability approaching 1). This provides us with an interval of s-values of size approximately $2m$ although in many cases these values of n will have small prime divisors and in some cases the resulting polynomial will itself be reducible.

We must now return to the question of size. For polynomials retaining a smaller Galois group the growth is confined to the constant term — and by skewing the sieve region the effect of this could be minimised. We should also recall that we may be able to improve on such a polynomial by performing translations. Whether it is reasonable to consider the extremes of this method as special cases may be open to debate however, the small size of the higher coefficients, the small Galois group and the possibility of useful translations do mark these polynomials out from the general case in which the skew is most usually far less severe and the Galois group is S_d.

Further to this we note that while additions to the higher coefficients may alter the Galois group and the field, the polynomials that result can still be extremely small.

Similar methods are available in the case of the other special forms. Once we

have a variant of the SNFS polynomial that is itself a base-m representation we are able to utilise translations in order to produce additional polynomials which define an isomorphic number field.

As we shall see Murphy's schema is capable of producing polynomials of these forms. Some simple cases may be noticed by considering n written in various bases and looking for patterns. For instance the special structure of $n = 2^r - 1$ is immediately exposed when n is written in binary. Hence we are interested in whether Murphy's schema can contribute any more than this. The ability to recognise the less obvious special cases — and perhaps see this as a continuum rather than a severe split into a handful of pleasant cases and the general case may enable us to quantify more precisely any threat to cryptographic methods from the polynomial selection algorithms.

7.4 Producing special case variants using Murphy's schema

The first step in both Murphy's skewed and non-skewed schema is to select leading coefficients which are divisible by the powers of many small primes. In an implementation we must define what we mean by "small" and we take this to be defined by choosing a factor base of small primes. We may then consider any leading coefficients in the range that have a divisor c that is smooth over this factor base, these divisors can be selected in a randomised manner. In addition we allow $c = 1$ to be chosen in this stage.

We then discard any polynomial that has particular coefficients that are not sufficiently small: in the non-skewed case we require all coefficients to be χ-small for some user selected bound χ; in the skewed case just the higher coefficients, usually a_d, a_{d-1}, a_{d-2}, must be small. In the special cases we are interested only in polynomials with small coefficients so this step will not discard any special form polynomial except due to matters of skew in the non-skewed case. In the non-skewed algorithm we then approximate $\alpha(F)$ and retain polynomials for which this is small. This step is problematic in the special cases since if we are to allow the selection of a small polynomial with Galois group strictly smaller than the full symmetric group we have seen that this value may not be completely appropriate

— either for comparing such polynomials or for comparing them with polynomials of a more general form. We must take into account both size and root properties simultaneously — as occurs in the skewed case where the ratings for the size and root properties are combined. There are a variety of ways of doing this.

In the skewed case we also calculate rotations and translations with an aim of minimising the size of the polynomial and forcing good root properties. Again, we must ensure that the size and root properties are both taken into account simultaneously.

Once we have compiled a list of "good" polynomials and respective m values we must compare them. In this case the metric used takes into account both size and root properties as described in chapter 3. Hence this method should cope well with the presence of general and special polynomials in the same list. Calculation of other such measures e.g. $\mathbb{E}$ can also be used.

Using our own interpretation of Murphy's non-skewed schema and Chris Monico's interpretation of Murphy's skewed schema [70] we have been able to produce lists of "good" polynomials containing SNFS-variant polynomials of the forms outlined in the previous section. These small polynomials with Galois group strictly smaller than the full symmetric group are produced on the input of various known special structure integers without providing any specific guidance regarding the known structure. The polynomials produced are not usually those that would be chosen by a human. It is an open question whether the method will always eventually induce this effect when presented with a number with a special structure that can be defined by a polynomial form.

In the case $n = r^e - s$ the output of such results appears to be a side effect of the first step of both of these schema in which we attempt to produce leading coefficients in the appropriate range which are divisible by many small p^a, p prime. There is nothing to suggest that the methods were explicitly designed to produce special case polynomials in these situations though it does correlate with the primary motives which were to find polynomials of small size with good root properties.

For instance, if $n = r^e - s$ with s of "reasonable" size (feasibly up to that discussed above) and r an integer which is smooth over the collection of small primes used in

this step then it is possible that the method will hit on an expansion in which the coefficients (excepting the constant term) are all divisible by r^a with $a = e - kd$ for some k. In addition the leading coefficient can be extremely small with respect to n (although clearly this depends on the size of r). This occurrence would produce a number field isomorphic to that produced by a human and hence the same Galois group to that produced by a human. Further translations or different values of k may produce "nicer" representations of the number field. If r is not prime then we do not necessarily require the involvement of all of the prime factors in the leading coefficient — complicating matters further.

We have also been able to produce translations of the polynomials chosen by humans in the cases in which we are working with cofactors of Cunningham numbers (defined by Aurifeuillian factorisations) and integers of various other forms. We give a variety of specific examples presently. It is difficult, if not impossible to determine the density of integers that will eventually induce this effect not least because the selection of leading coefficients is randomised. It is of course, impossible to say whether Murphy's schema are capable of isolating other special cases that have not yet been noted. On the other hand, this does raise an interesting philosophical question regarding the number field sieve.

The special number field sieve has, to date, been considered to be a collection of factorisations in which a human selected a polynomial, with reference to known special structure, in order to produce a favourable runtime based on the small size of that polynomial. The work of Murphy and others has firmly established that it is not just the size but also the root properties that are of importance, further to this we can recognise (subsets of) the current collection of special cases as having distinct characteristics, as well as small size, that may plausibly allow an advantage.

The general case relies on automated methods assumed to produce, in some sense, "random" polynomials although attempts are made to force good size and root properties. Since it appears that these general methods and other automated processes can, in certain circumstances, produce the same number field as a human and in others identify structure of which a human may have been previously unaware it appears that the divide between special and general may be somewhat blurred. This brings us to the pertinent question:

Question 3 *To what extent is it possible to utilise Murphy's schema or any other*

automated search to produce a special case polynomial without reference to any special structure that might exist? Put another way, can we check automatically for (some types of) "specialness"?

7.4.1 In practice

We will consider some specific examples of this phenomena and note how the production of these SNFS variants arises.

If we have a number with structure $r^e - s$ with r a small integer (either prime or a product of small prime powers) and $|s| < n^{(1/d)}$ then the first step in the schema may eventually hit on a leading coefficient $a_d = r^{e-kd}$ for some k, if this occurs then the structure is exposed. For instance, in the case of the factorisation of F_9 with $d = 5$ we may produce the polynomial

$$f_{\text{auto}}(X) = 2^2X^5 + (2^2 \cdot 5)X^4 + (2^3 \cdot 5)X^3 + (2^3 \cdot 5)X^2 + (2^2 \cdot 5)X + 5$$

of Galois group $F(20)$. This is in fact the polynomial $f_{\text{var}\,1}(X) = 4X^5 + 1$ evaluated at $X + 1$ and hence is a translation of the first variant of the SNFS original. The difference in the coefficient size between the automatically generated polynomial and the (variant of the) SNFS original may not be of any relevance. In the above example the SNFS original polynomial is $f(X) = X^5+8$ with root $m = 2^{103}$ modulo F_9; f_{var} has root $m/2$ modulo F_9 and hence f_{auto} has root $m/2-1$ modulo F_9. Hence, f is $1/m$-small, f_{var} is $8/m$-small and f_{auto} is $40/(m/2-1) \approx 80/m$-small. These numbers are so small that the difference is irrelevant.

Even if we input a degree other than the one chosen by a human as appropriate we still expose the structure:

$$16X^4 + 64X^3 + 96X^2 + 64X + 17, \;\; \text{Gal} = E(4)$$
$$4X^6 + 24X^5 + 60X^4 + 80X^3 + 60X^2 + 24X + 5, \;\; \text{Gal} = D(6)$$

Since these polynomials are again translations of SNFS variants we find that they exhibit a small Galois group.

We note that this effect is at least possible regardless of the degree chosen (pre-

suming that the degree is less than e and that a relevant value of k exists) and is in no way dependent on using the degree suggested by the SNFS asymptotics — we will still produce a small polynomial that has structure in the coefficients leading to a Galois group smaller than the full symmetric group.

This is a fairly simple example — one that could have been noted by preceding the schema by examining the base-p expansion of n for each prime in our set of small primes. However, the same method can allow us to use the underlying structure in a number with a base that is only a product of prime powers in the factor base. This would have been harder, though not impossible, for a human to notice. For instance the example $35^{97} + 1$ produces the polynomial

$$1225X^5 + 6125X^4 + 12250X^3 + 12250X^2 + 6125X + 1226$$

with Galois group $F(5)$. This is a translation of the first SNFS variant.

We can increase the complexity of the base, as long as it remains smooth over the "small" primes we may be able to disclose it — due to the random nature of the schema we may of course never hit on any of the values of a_d that would produce a "nice" polynomial. However, once we have just one such polynomial with recognisable characteristics we can often form a more beneficial representation of n and the number field by hand.

Some "larger" examples, all of these were produced in just a matter of minutes search time

1.	$1859^{51} + 68$ $1331X^6 + 7986X^5 + 19965X^4 + 26620X^3 + 19965X^2 + 7986X + 1399$
2.	$2299^{49} + 100$ $2299X^6 + 13794X^5 + 34485X^4 + 45980X^3 + 34485X^2 + 13794X + 2399$
3.	$104329^{33} + 994$ $X^6 + 6X^5 + 15X^4 + 20X^3 + 15X^2 + 6X + 995$
4.	$8303^{43} + 14526$ $8303X^6 + 49818X^5 + 124545X^4 + 166060X^3 + 124545X^2$ $+49818X + 22829$
5.	$17017^{37} + 736272$ $17017X^6 + 102102X^5 + 255255X^4 + 340340X^3 + 255255X^2$ $+102102X + 753289$

For instance in the case of integer 2 an example of a rival polynomial produced by the method is

$$\begin{aligned}
&1336778170X^6 + 1670894748656X^5 + 106566111132040465357 5X^4 \\
&-38346684967100920511069649X^3 - 28374058298255124518418741 12X^2 \\
&\qquad +37929243856754580688788768187 16X \\
&\qquad +56463350327426454808855425218 4143
\end{aligned}$$

and in the case of integer 5:

$$\begin{aligned}
&22048670826X^6 + 2854864650807X^5 - 31596247306722303875X^4 \\
&+56971499294395915097864 3X^3 + 52083723445675089164964 19X^2 \\
&-51473460724055634558567025 6X - 78330819395096110133251 35755
\end{aligned}$$

both with Galois group S_6. We did not experiment with leaving the polynomial selection code running for longer periods of time and with more difficult examples but shall leave a more full investigation into the actual capabilities of the method over time for future work.

For numbers of the form $(r^e + 1)/(r^k + 1)$ the above situation does not occur, in fact the method appears to settle on $a_d = 1$ in those examples we have run. Again, the polynomials produced share a Galois group $C(6)$ with the "natural" polynomial produced by a human with knowledge of the inherent structure. It is presumably the small size of the polynomial that allows it to be selected (and the fact that we allow $a_d = 1$ at all — the rotations and translations appear to have no effect in the cases we have tried). An example of this is $(17^{119} + 1)/(17^{17} + 1)$ for which the polynomial

$$X^6 + 11X^5 + 51X^4 + 127X^3 + 179X^2 + 135X + 43$$

is selected. The natural polynomial produced for this number by a human would be

$$X^6 - X^5 + X^4 - X^3 + X^2 - X + 1.$$

That produced by Murphy's schema is a translation ($X \mapsto X + 2$) of this polynomial and hence defines an isomorphic number field.

We can also achieve good results for cofactors of Cunningham numbers that are

produced using Aurifeuillian factorisations, the number $3^h + 3^{\frac{h+1}{2}} + 1$, $h = 331$ (which suggests the polynomial form $X^6 + 3X^3 + 3$) produces:

$$3X^6 + 18X^5 + 45X^4 + 63X^3 + 54X^2 + 27X + 7$$

This polynomial is a translation of $3X^6 + 3X^3 + 1$ which produces an isomorphic field.

Of course, we could add to the implementation a deterministic check that would betray the more simplistic examples almost immediately. It is really the examples where the structure is not immediately clear from a collection of base-p expansions in which we would be interested.

Question 4 *Can we automatically isolate more difficult examples which we cannot uncover by other means?*

7.5 Polynomial selection and RSA

It seems unlikely that a single method can reliably make use of all special structure that could be present however, the ability to automatically take advantage of an unknown but present special structure in even a sparsely distributed set of integers seems to blur the line between special and general. The existence of a single algorithm — simplistic or no — that can, on occasion, produce such results could conceivably hold implications for the RSA cryptosystem [84] unless we take the existence of the algorithm into account and ensure that it cannot pose a threat.

In the introduction to this thesis we gave a brief outline of the RSA algorithm. As well as enabling secure communication between two individuals that have not met or exchanged secret keys some public-key cryptosystems also enable individuals to produce a digital signature. RSA is one such system, we will briefly explain how this is accomplished and introduce some other necessary facts about the system. More information can be found in [8, 68, 84].

A digital signature of a file or message depends on the contents of the message

itself and on knowledge of the private key used to "sign" it — in this case the private part of the RSA key pair. The RSA cryptosystem naturally gives rise to a signature scheme as the operation of "encryption" is bijective. In simplistic terms, we can "encrypt" and "decrypt" with either a public key or a private key. If we encrypt with the someone else's public key we may then communicate securely with that person. If, on the other hand, we "encrypt" with our own private key we are able to form a piece of information or signature that can be verified as being created by us easily (by "decrypting" with our public key) and, most importantly, such that (it is believed) no other person could have created the signature without knowledge of our private key.

We create $n = pq$, p, q prime, our private key (n, d), our public key (n, e) as in chapter 1. However, to sign a block of plaintext m we "encrypt" with our private key (for simplicity we assume the message $m < n$; $m \geq n$ complicates notation but not implementation):

$$s \equiv m^d \bmod n$$

to obtain s, a signature for the message m.

If someone should wish to verify that we signed the message they "decrypt" with our public key. We noted in the introduction that access to the factorisation of $n = pq$ results in a complete break of the cryptosystem — this includes the signature scheme. There are a myriad of other issues of significant importance that must be addressed in order to use the ideas above to create a secure signature scheme, however, most of these will not impact on the subsequent discussion. In fact we will not talk about the technical details of the algorithm except for how the primes p and q are selected.

Prime selection for RSA

When selecting primes for RSA we may choose to adhere to some or all of the following [8, 68]:

- p and q should be approximately the same size (bitlength) and of a size large enough so as to preclude the possibility of factoring using the elliptic curve factoring algorithm.

- p and q should not be too close together, if they are then $p, q \approx \sqrt{n}$. Such n are considerably easier to factor than a general number of the same size as they are immediately susceptible to Fermat's factorisation method. If p, q are chosen randomly this problem is side stepped as we would have $p, q \approx \sqrt{n}$ with extremely small probability.

- It is possible that we might choose to use strong primes. Strong primes are p and q chosen such that:

  1. In order to avoid attack by the Pollard $p-1$ factoring algorithm we ensure that $p-1$ has a large prime factor, say r.
  2. In order to avoid the Pollard $p+1$ factoring algorithm we ensure that $p+1$ has a large prime factor.
  3. To avoid so called "cycling attacks" $r-1$ is chosen to have a large prime factor.

  However, while there is little additional cost to the use of strong primes it is not certain they add any security. As discussed in [68, note 8.8] if p and q are sufficiently large, randomly chosen primes then we would expect $p \pm 1$, $q \pm 1$ to have large prime factors in any case — and a cycling attack to have only a negligible probability of success. In addition, strong primes offer no protection against the elliptic curve factoring algorithm (a generalisation of the $p-1$ and $p+1$ attacks).

If the above method of prime selection does not preclude the use of a modulus n with some special form which cannot be easily identified by a human yet can be isolated as having a special form for the number field sieve by a schema such as Murphy's then the ability to automatically isolate special forms may have implications for RSA.

Discussion

Obviously we require any cryptosystem to be "secure" but there are various models of security. Computational security measures the amount of computational effort which would be required to defeat a system using current technology and methods. Many cryptosystems are only considered to be computationally secure, for an overview of this and other models of security see [68].

RSA is generally considered to be computationally secure if current guidelines for parameter selection are followed. In particular, guidelines on the length of n that is currently considered to give short, medium and long term security are produced based on estimates of the size of number that the fastest general factoring algorithm will be able to factor given a specific set of resources. These estimates are based on knowledge of the current record factorisations and on the assumption that no new, faster, general algorithm will be invented.

Non-repudiation is another commonly required feature of public-key/signature schemes. It is defined to be any aspect of a system or service that allows us to prevent the denial of previous commitments or actions [68].

Clearly, public-key systems and digital signatures have no worth if we are able to successfully claim that we were not involved in an action when we are in fact responsible. This can be quite a subtle problem and requires various key management techniques some of which have significant overheads. For a general overview see [68]. Ideally we would like to avoid or remove as many opportunities as possible for an individual to successfully repudiate their actions.

The security of RSA lies in the secrecy of the private key (n, d). As noted earlier, factorisation of n leads to a complete break of the system and thus a weak key for RSA is any private key for which n can be factored, with current methods, in polynomial time. The above rules for selecting p and q ensure that such n are very rare. However, we might consider an RSA key to be a *computationally weaker key* if the cost to factor n, while not polynomial, is substantially less than assumed. For instance, a modulus n might be selected so that it is out of the current range of GNFS but should n have a special form which allows a parametrisation with SNFS characteristics we may find that n then falls into the current range for SNFS. Thus the key would be computationally weaker than intended.

If we assume that an individual had by some circumstance such a computationally weaker key then the existence of automatic methods which may be able to isolate the parameters for the special case without prior knowledge of the structure leaves the individual open to an opportunistic attack. However, the cost of factoring the modulus would still be very high.

Perhaps of more interest is the possibility of a repudiation attack. If an assailant

could produce in some manner a computationally weaker key of this nature that appeared to obey the usual selection criteria for p and q above they could then use this to produce electronic signatures with the intention of later repudiating their actions.

The assailant would not have to produce a factored modulus in order to repudiate their actions, they would only need to show, by a method which does not use knowledge of the structure present, that the modulus is weaker than they intended and within the current range of factorisation. In this case the burden of proof that the key was created to be weaker would lie with an arbitration authority.

The risk here is not only that there are plausibly weaker keys we may currently allow to be used but the method by which these particular weaker keys can be recognised. The concerns are as follows: firstly, we do not have any immediate way of quantifying the density of keys which have "nice" polynomials nor of reliably testing for all special forms that could exist; secondly, that keys of this form could be automatically shown to be weaker than intended *without producing a factored modulus or evidence that such a factorisation has occurred.* In fact a "weaker" modulus n can still be very expensive to factor, it is determined to be weaker by the fact that, given current technology and a specific monetary input, it can be factored but general numbers of the same size as n cannot be.

Even supposing that a factorisation of n would show that the assailant had purposely undermined their own RSA key (for instance, if p and q did not fulfil the usual criteria or could be shown to be selected at random (as a pair) with very small probability) the cost of factoring n is high and may not prove to be a cost effective argument for an arbitration authority. In addition, if the form of the special case is one that has not been well investigated the argument that a modulus which can be written in this polynomial form will occur with very low probability may not be valid.

Question 5 *Does there exist a method to produce, at will and in a manner which is difficult to detect, computationally weaker keys of the nature outlined above?*

In order to avoid such repudiation and opportunistic attacks on RSA we must therefore consider the probability of primes leading to any special form being

selected at random and used in an RSA key. If the likelihood of this occurring is high enough, we may consider whether we can check for any "special" cases currently known in order to ensure that weaker keys are not used. It may be useful to run Murphy's scheme on any modulus n the factorisation of which is just infeasible when considered as a general number but which could lie in the reach of SNFS. However, Murphy's scheme for selecting polynomials has a randomised component so while we can test any modulus n the inability to find a pleasant polynomial form does not ensure that one does not exist. In addition to this the goal posts may not be static — it is unclear whether there may be other special forms of the number field sieve yet to be discovered.

Question 6 *Do there exist many special forms of the number field sieve or can numbers be seen on a continuum with no clear divide between a "difficult" general class of numbers and a "nice" special class of numbers?*

It is difficult, if not impossible, and certainly beyond the scope of this thesis to quantify all "nice" polynomial forms a schema such as Murphy's can identify. It is therefore not possible at this time to give an in depth analysis of the likelihood of producing a weaker RSA key than was intended though at first glance it would seem these keys would either be rare or lie on a continuum of NFS "hardness". In addition to this any argument would necessarily take into account only the abilities of published schema for producing polynomials. It seems unlikely that any one scheme can isolate all special forms that result in a reduced run time in the number field sieve and of course any number of automated methods could be formed in future to find specific polynomial forms without knowledge of the structure of a number. For these reasons it does not seem possible to quantify the risk of using a weaker key than was intended nor the risk that an individual may purposely use a weaker key with the intention of later repudiating their actions. We leave such considerations to further study.

7.6 Summary

We have noted an apparent blurring in the distinction between the special number field sieve and the general number field sieve. In particular we have seen that

a variety of special forms can be recognised by Murphy's schema without direct access to the special structure present.

Further to this, we have considered possible implications for RSA noting that such automated methods of isolating special forms can recognise keys which are weaker than intended. We hope but cannot conclude that this will occur only in rare circumstances if we assume randomness but also note the possibility of an attacker selecting such a key with the intention of later repudiating their actions. In this case the attacker has choice and the authority the burden of proof that the key was specifically chosen.

Finally, with regards to any additional method that can be used to speed up cases of the number field sieve which use special structure we are now in a situation where we must consider implications for special structure numbers that have not been recognised as such by a human and may have been used for cryptographic purposes.

Chapter 8

Summary

8.1 Further work

The method to estimate the quantity of relations produced by the classical sieve cannot be used as a stand alone method for comparing parametrisations of a factorisation or variants of the number field sieve as we cannot be sure that the method is stable. In addition the technique is not applicable in the case of the lattice sieve.

More work could be done to test the technique of splitting the sieve region. Should the resources be available we may wish to enter into large scale tests in order to verify the stability of the estimate. This would be all the more beneficial if the technique or a similar method could be applied to other sieving mechanisms. We might also consider using the outcome of the investigation into the underestimate as a starting point to developing a more natural and adaptive method of estimation.

The open questions posed in chapter 7 provide obvious directions for further work that is clearly beyond the scope of this thesis. However these questions are unlikely to be answered with ease. A more prudent course of research would be to consider what other special forms may be factored with a significantly reduced runtime in comparison with the general case and derive methods to automatically isolate these. A more general aim would be to work towards a

polynomial selection algorithm that could somehow be used as a judge for the likely difficulty of factoring the number provided as input.

In the case of the subfield structure we would aim to find other ways of using the structure that did not involve a linear side and hence that would allow us to tap into the promising source of relations that the subfields appear to provide. We may also consider working in extensions of a main field rather than in subfields of one.

8.2 In summary

We have investigated the source of the under estimates in a method for estimating the quantity of relations produced by the number field sieve and provided evidence that this is not in fact due solely to the range of the size of values taken by the polynomial but an effect generated by the skew of the values. We have considered splitting methods rooted in this idea and that of splitting the sieve region into equal sized subregions. We noted some of the negative qualities of the method of splitting the region equally — most specifically the problem of deciding the quantity of subregions that would be most applicable to any given problem. We provide evidence that another method which does not have this flaw can produce reasonable estimates.

We have established some of the characteristics of the special cases of the number field sieve so that we might define "specialness" via a set of "nice" properties, rather than via the idea that the polynomials are created using known special structure to be particularly small. We have contrasted these with the general case and found that care should be taken when comparing general and special cases as the underlying factor base structure is, on average, quite different.

In particular we have noted that in the case that the algebraic number field has composite degree and a Galois group strictly smaller than the full symmetric group it is possible that we may have subfield structure. We have seen that in the case of $d = 4$ and $d = 6$ in some of the main special cases produced by humans subfield structure is present.

We considered the most obvious and natural method of utilising the subfield structure found in the main special cases. We show how the number field sieve can be adapted to make use of the subfields and find that this can have a significant effect on the algebraic side. However, the ensuing explosion in size of the auxiliary numbers that we hope to be smooth on the linear side leads us to the conclusion that this method is not of practical use.

We have considered the special cases in the context of Murphy's general polynomial selection schema posing some open questions regarding this. We note that for certain special cases while Murphy's schema are not generally able to produce the polynomials that a human might produce that the schema is capable of producing an isomorphic number field, often with a small representation, with no guidance as to the special structure present. We note that there are other ways to automatically check for some of the special numbers which might be incorporated as a first step in the schema.

This would appear to blur the distinction between the special and general cases of the number field sieve. It is difficult, if not impossible to know the density of "special" forms that exist or that could be automatically produced without prior knowledge of any special structure. We might now wish to consider variants of the number field sieve that are only applicable in the special cases as it is possible (though we hope unlikely) that a number used for cryptographic purposes could be recognised automatically as a special case without human intervention. Finally, the existence of automated methods by which we can isolate some forms of "specialness" without knowledge of any structure in the number to be factored raises the question of the possibility of repudiation or opportunistic attacks, which we assume but cannot conclude are rare, on the RSA cryptosystem.

Appendix A

SNFS factorisations

We collect together some example SNFS factorisations from the literature including the record breaking factorisations from 1997 onwards. In the most part these factorisations are here because we have information regarding F the polynomial used and hence we may calculate the Galois group Gal(F) and an approximation to $\alpha(F)$. In all cases F has small coefficients, a small Galois group and in most cases a positive value of $\alpha(F)$. We can also see that the range of $\alpha(F)$ values is quite narrow.

In the table $x, y+$ denotes $x^y + 1$ and $x, y-$ denotes $x^y - 1$. An entry such as $C145$ fr. $2,488+$ refers to the cofactor with 145 digits from $2^{488} + 1$.

The approximation to $\alpha(F)$ is calculated in a similar manner as the calculation of $\alpha(f)$ as described in chapter 4. We allow random sampling across $[-10^4, 10^4] \times [0, 10^4]$ and use a factor base bound of 10^4 (smaller factors almost entirely control the value).

Table A.1: Table of SNFS factorisations from the literature including Galois group and $\alpha(F)$

Number	Ref.	Polynomial f	Gal(F)	$\alpha(F)$
$2,28+$	[80]	$X^3 + 2$	S_3	1.58
$2,512+$	[62]	$X^5 + 8$	$F(20)$	0.97
$3,239-$	[63]	$X^5 - 3$	$F(20)$	1.15

continued on next page

Table A.1: *continued*

Number	Ref.	Polynomial f	Gal(F)	$\alpha(F)$
$2,373+$		X^5+4	$F(20)$	1.20
$7,149+$		X^5+7	$F(20)$	1.03
$2,457+$		X^5+8	$F(20)$	0.97
$C145$ fr. $2,488+$	[6]	X^5+4	$F(20)$	1.20
$C151$ fr. $2,503+$		$8X^5+1$	$F(20)$	0.97
$2,523-$		$8X^5-1$	$F(20)$	0.98
$C123$ fr. $2,511-$	[39]	$X^6-10X^4+24X^2-8$	$C(6)$	1.92
$C162$ fr. $12,151-$	[49]	$12X^5-1$	$F(20)$	0.43
$C98$ fr. $7^{128}+6^{128}$		X^4+1	$E(4)$	2.66
$C106$ fr. $2,543-$		$4X^4+2X^2+1$	$E(4)$	1.74
$C119$ fr. $3,319-$		$X^5+X^4-4X^3-3X^2+3X+1$	$C(5)$	2.32
$C135$ fr. $73,73+$		X^5+73^2	$F(20)$	0.76
$6,199-$	[42]	X^5-6	$F(20)$	0.93
$10,158+$		$8X^5+25$	$F(20)$	-0.36
$C144$ fr. $7,187-$		$X^5+X^4-4X^3-3X^2+3X+1$	$C(5)$	2.32
$C156$ fr. $2,559-$		$X^6+X^5-5X^4-4X^3$ $+6X^2+3X-1$	$C(6)$	3.15
$5,505-$		$25X^4+25X^3+15X^2+5X+1$	$C(4)$	1.44
$10,81+$	[85]	$10X^5+1$	$F(20)$	0.35
$10,184+$		X^5+10	$F(20)$	0.36
$10,91+$		$10X^5+1$	$F(20)$	0.35
$10,194+$		X^5+10	$F(20)$	0.36
$10,197-$		X^6-10	$D(6)$	1.33
$6,256+$		$6X^5+1$	$F(20)$	0.94
$2,587+$		$4X^5+1$	$F(20)$	1.19
$2,617+$		$4X^5+1$	$F(20)$	1.19
$2,619+$		X^5+2	$F(20)$	1.43
$5,257+$		$25X^5+1$	$F(20)$	0.53
$35,97+$		$1225X^5+1$	$F(20)$	1.01
$97,73-$		$6X^5-9409$	$F(20)$	1.46
$12,167+$		$144X^5+1$	$F(20)$	-0.32
$4^{232}+3^{232}$		$16X^5+9$	$F(20)$	0.30
$2,751-$	[37]	$2X^6-1$	$D(6)$	2.47
$668\times 2^{668}-1$	[65]	$-4X^6+167$	$D(6)$	2.05
$6,257-$		X^6-6	$D(6)$	1.91
$5,289+$		X^6+5	$D(6)$	1.71

continued on next page

Table A.1: *continued*

Number	Ref.	Polynomial f	Gal(F)	$\alpha(F)$
$11,197+$		X^6+11	$D(6)$	1.28
$2,673-$		$2X^6-1$	$D(6)$	2.47
$12,178+$		$4X^6+9$	$D(6)$	0.58
$5,298+$		X^6+25	$D(6)$	1.43
$12,197-$		X^6-12	$D(6)$	1.37
$10,227-$		X^6-10	$D(6)$	1.33
$2,713-$		X^6-2	$D(6)$	2.47
$2,757-$		$2X^6-1$	$D(6)$	2.47
$3,491+$		X^6+3	$D_6(6)$	2.68
SNFS records				
$C180$ fr. $12,167+$	[72]	X^5-144	$F(20)$	-0.32
$(2^{15}-135),41-$	[19]	$X^5-(2^{15}-135)$	$F(20)$	0.63
$C211$ fr. $10,211-$	[16]	$10X^6-1$	$D(6)$	1.33
$2,773+$	[15]	X^6+2	$D(6)$	1.94
$2,809-$	[44]	Not reported	-	-
$2,1642M$	[2]	X^6+2X^3+2	$D(6)$	2.23
$C274$ fr. $6,353-$	[3]	X^6-6	$D(6)$	1.91

References

[1] L. M. Adleman. Factoring Numbers using Singular Integers. In *Proc. 23rd Annual ACM Symp. on Theory of Computing (STOC)*, pages 64–71. ACM Press, 1991.

[2] K. Aoki, Y. Kida, T. Shimmoyama, Y. Sonoda, and H. Ueda. 248-digit SNFS factorization. `http://www.crypt-world.com/announcements/SNFS248.txt`, 2004. (Aug. 2006).

[3] K. Aoki, Y. Kida, T. Shimoyama, and H. Ueda. 274-digit SNFS factorization. `http://www.crypto-world.com/announcements/SNFS272.txt`, 2006. (Aug. 2006).

[4] E. Bach and R. Peralta. Asymptotic Semismoothness Probabilities. *Mathematics of Computation*, 65:1701–1715, 1996.

[5] E. A. Bender and E. R. Canfield. An Approximate Probabilistic Model for Structured Gaussian Elmination. *J. Algorithms*, 31:271–290, 1999.

[6] D. J. Bernstein and A. K. Lenstra. *A General Number Field Sieve Implementation*, pages 103–126. In Lenstra and Lenstra [61], 1st edition, 1993.

[7] H. Boender. The Number of Relations in the Quadratic Sieve Algorithm. Technical Report NM-R9622, CWI, Amsterdam, 1996. Chapter 4, Phd Thesis, University of Leiden, `http://ftp.cwi.nl/CWIreports/NW/NM-R9622.pdf` (Aug. 2006).

[8] D. Boneh. Twenty Years of Attacks on the RSA Cryptosystem. *Notices of the AMS*, 46:203–231, 1999.

[9] R. P. Brent. Recent Progress and Prospects for Integer Factorisation Algorithms. In *Computing and Combinatorics (Sydney, 2000)*, volume 1858 of *LNCS*, pages 3–22. Springer-Verlag, 2000.

[10] R. P. Brent, P. L. Montgomery, and H. J. J. te Riele. Factorizations of Cunningham numbers with bases 13 to 99: millennium edition. Technical Report MAS-R0107, CWI, Amsterdam, 2001. Available at `http://ftp.cwi.nl/CWIreports/MAS/MAS-R0107.pdf` (Aug. 2006).

[11] J. Brillhart, D. H. Lehmer, L. Selfridge, B. Tuckerman, and S. S. Wagstaff, Jr. *Factorizations of $b^n \pm 1$, $b = 2, 3, 5, 6, 7, 10, 11, 12$ up to higher powers.* Amer. Math. Soc., Providence, RI, 3rd edition, 2002. Available at `http://www.ams.org/online_bks/conm22/` (Aug. 2006).

[12] J. P. Buhler, H. W. Lenstra, Jr., and C. Pomerance. *Factoring Integers with the Number Field Sieve*, pages 50–94. In Lenstra and Lenstra [61], 1st edition, 1993.

[13] S. Cavallar. Strategies in Filtering in the Number Field Sieve. In *Algorithmic Number Theory, ANTS-IV*, volume 1838 of *LNCS*, pages 209–231. Springer-Verlag, 2000.

[14] S. Cavallar. The Three-Large-Primes Variant of the Number Field Sieve. Technical Report MAS-R0219, CWI, Amsterdam, 2002. Available at `http://www.cwi.nl/ftp/CWIreports/MAS/MAS-R0219.pdf` (Aug. 2006).

[15] S. Cavallar, B. Dodson, J. Fougeron, J. Gilchrist, A. Lenstra, P. Leyland, W. Lioen, P. Montgomery, A. Muffett, NFSNET2 (M. Bruestle, S. Contini, P. Dodson, E. C. Dost, S. Edick, T. Holroyd, J. Klos), and H. te Riele. 233-digit SNFS factorization. `http://ftp.cwi.nl/pub/herman/SNFSrecords/SNFS-233`, 2000. (Aug. 2006).

[16] S. Cavallar, B. Dodson, A. Lenstra, P. Leyland, W. Lioen, P. Montgomery, H. te Riele, and P. Zimmermann. 211-digit SNFS factorization. `http://ftp.cwi.nl/herman/SNFSrecords/SNFS-211`, 1999. (Aug. 2006).

[17] S. Cavallar, B. Dodson, A. K. Lenstra, P. C. Leyland, W. M. Lioen, P. L. Montgomery, B. Murphy, H. J. J. te Riele, and P. Zimmermann. Factorization of RSA-140 using the Number Field Sieve. Technical Report MAS-R9925, CWI, Amesterdam, 1999. Available at `http://www.cwi.nl/ftp/CWIreports/MAS/MAS-R9925.pdf` (Aug. 2006).

[18] S. Cavallar, W. M. Lioen, H. J. J. te Riele, B. Dodson, A. K. Lenstra, P. L. Montgomery, B. Murphy, and et al. Factorization of a 512-bit RSA Modulus. Technical Report MAS-R0007, CWI, Amsterdam, 2000. Available at `http://www.cwi.nl/ftp/CWIreports/MAS/MAS-R0007.pdf` (Aug. 2006).

[19] S. Cavallar, P. Montgomery, and H. te Riele. 186-digit SNFS factorization. `http://ftp.cwi.nl/pub/herman/SNFSrecords/SNFS-186`, 1998. (Aug. 2006).

[20] S. H. Cavallar. *On the Number Field Sieve Integer Factorisation Algorithm.* PhD thesis, Leiden University, 2002. Available at `http://www.cwi.nl/ftp/herman/theses/Stefania.ps.Z` (Aug. 2006).

[21] H. Cohen. *A Course in Computational Algebraic Number Theory.* Springer-Verlag, Berlin, Germany, 1st edition, 1993.

[22] H. Cohen. *Advanced Topics in Computational Number Theory.* Springer-Verlag, New York, USA, 1st edition, 2000.

[23] H. Cohen, F. Diaz y Diaz, and M. Olivier. Counting Discriminants of Number Fields of Degree up to Four. In *Algorithmic Number Theory, ANTS-IV*, volume 1838 of *LNCS*, pages 269–283. Springer-Verlag, 2000.

[24] S. Contini. Factoring Integers with the Self Initialising Quadratic Sieve. Master's thesis, University of Georgia, 1997. Available at `http://www.crypto-world.com/Contini.html` (Aug. 2006).

[25] J. H. Conway, A. Hulpke, and J. McKay. On Transitive Permutation Groups. *LMS J. Comput. Math.(electronic)*, 1:1–8, 1998.

[26] D. Coppersmith. Modifications to the Number Field Sieve. *J. Cryptology*, 6:169–180, 1993.

[27] D. Coppersmith. Solving Linear Equations over $GF(2)$: Block Lanzos Algorithm. *Linear Algebra and its Applications*, 192:33–60, 1993.

[28] D. Coppersmith. Solving Homogeneous Linear Equations over $GF(2)$ via Block Wiedemann. *Mathematics of Computation*, 62:333–350, 1994.

[29] J.-M. Coveignes. Computing a Square Root for the Number Field Sieve. In *[61]*, pages 95–102.

[30] P. A. Crouch and J. H. Davenport. Lattice Attacks on RSA-encrypted IP and TCP. In *Cryptography and Coding 2001*, volume 2260 of *LNCS*, pages 329–338. Springer-Verlag, 2001.

[31] A. J. C. Cunningham and H. J. Woodall. *Factorisation of* $(y^n \pm 1)$, $y = 2, 3, 5, 6, 7, 10, 11, 12$ *Up to High Powers (n).* Hodgson, London, 1925.

[32] M. Daberkow, C. Fieker, J. Klüners, M. Pohst, K. Roegner, and K. Wildanger. KANT V4. *J. Symbolic Comp.*, 24:267–283, 1997.

[33] H. Davenport. *The Higher Arithmetic.* Cambridge University Press, Cambridge, United Kingdom, 7th edition, 1999.

[34] N. G. de Bruijn. On the Number of Positive Integers $\leq x$ and Free of Prime Numbers $> y$. In *Nederl. Acad. Wetensch. Proc.*, volume 54 of *Ser. A.*, pages 50–60, 1951.

[35] W. Diffie and M. E. Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, 1976.

[36] J. Dixon. Computing Subfields in Algebraic Number Fields. *J. Austral. Math. Soc.*, 49:434–448, 1990.

[37] B. Dodson, J. Franke, A. K. Lenstra, P. Leyland, P. Montgomery, and H. te Riele. 227-digit SNFS factorization. `http://ftp.cwi.nl/pub/herman/SNFSgiants/SNFS-227`, 2002. (Aug. 2006).

[38] B. Dodson and A. K. Lenstra. NFS with Four Large Primes: an Explosive Experiment. In *Advances in Cryptology, CRYPTO '95*, volume 963 of *LNCS*, pages 372–385. Springer-Verlag, Berlin, 1995.

[39] M. Elkenbracht-Huizing. An Implementation of the Number Field Sieve. *Experimental Mathematics*, 5(3):99–114, 1996.

[40] M. Elkenbracht-Huizing. A Multiple Polynomial General Number Field Sieve. In *Algorithmic Number Theory, ANTS II*, volume 1122 of *LNCS*, pages 99–114. Springer-Verlag, 1996.

[41] R-M. Elkenbracht-Huizing. *Factoring Integers with the Number Field Sieve.* PhD thesis, Leiden University, 1997.

[42] R.-M. Elkenbracht-Huizing, P. L. Montgomery, R. D. Silverman, R. K. Wackerbarth, and S. S. Wagstaff. The Number Field Sieve on Many Computers. In *Number theory*, volume 19 of *CRM Proc. Lecture Notes*, pages 81–85. Amer. Math. Soc., Providence, 1999.

[43] L. Euler. Observationes de Theoremate quodam Fermatiano Aliisque ad Numeros Primos Spectantibus. *Comm. Acad. Sci. Petropol.*, 6:103–107, 1732/1733 published 1738. Available at `http://math.dartmouth.edu/`

~euler/docs/originals/E026.pdf (Aug. 2006); Translation available at http://math.dartmouth.edu/~euler/pages/E026.html (Aug. 2006).

[44] J. Franke, T. Kleinjung, F. Bahr, and P. Montgomery. 244-digit SNFS factorization. http://www.crypto-world.com/announcements/m809.txt, 2003. (Aug. 2006).

[45] P. X. Gallagher. The Large Sieve and Probabilistic Galois Theory. In *Analytic Number Theory*, volume 24 of *Proc. Symp. in Pure Math.*, pages 91–101. AMS, Providence, 1973.

[46] J. E. Gower. Rotations and Translations of Number Field Sieve Polynomials. In *Advances in Cryptology, Asiacrypt 03*, volume 2894 of *LNCS*, pages 302–310. Springer-Verlag, 2003.

[47] H. W. Lenstra Jr. Factoring Integers with Elliptic Curves. *Ann. of Math.*, 126:649–673, 1987.

[48] A. Hildebrand and G. Tenenbaum. On Integers Free of Large Prime Factors. *Transactions of the American Mathematical Society*, 296:265–290, 1986.

[49] R. M. Huizing. An Implementation of the Number Field Sieve. Technical Report NM-R9511, CWI, Amsterdam, 1995. Available at http://www.cwi.nl/ftp/CWIreports/NW/NM-R9511.pdf (Aug. 2006).

[50] A. Hulpke. Block Systems of a Galois Group. *Experimental Mathematics*, 4:1–9, 1995.

[51] S. Hunter and J. Sorenson. Approximating the Number of Integers Free of Large Prime Factors. *Mathematics of Computation*, 66:1729–1741, 1997.

[52] E. Kaltofen. On Wiedemann's Method of Solving Sparse Linear Systems. In *Proc. AAECC 9*, volume 539 of *LNCS*, pages 29–38. Springer, 1991.

[53] J. Klüners and M. Pohst. On Computing Subfields. *J. Symbolic Computation*, 24:385–397, 1997.

[54] D. E. Knuth. *The Art of Programming: Semi-Numerical Algorithms.* Addison-Wesley, Reading, Massachusetts, 2nd edition, 1981.

[55] J. C. Lagarias and A. M. Odlyzko. Effective Versions of the Chebotarev Density Theorem. In *Algebraic number fields: L-functions and Galois properties*, Proc. Sympos., Univ. Durham, pages 409–464. Academic Press, London, 1977.

[56] R. Lambert. *Computational Aspects of Discrete Logarithms.* PhD thesis, University of Waterloo, 1996.

[57] C. Lanczos. Solution of Systems of Linear Equations by Minimized Iterations. *J. Res. Nat. Bureau Standards*, 49:33–53, 1952.

[58] S. Lang. *Algebraic Number Theory.* Springer-Verlag, 1st edition, 1986.

[59] S. Lang. *Algebra.* Addison-Wesley Publishing Company, USA, 3rd edition, 1993.

[60] D. Lazard and A. Valibouze. Computing Subfields: Reverse of the Primitive Element Problem. In *Computational algebraic geometry (Nice, 1992)*, volume 109 of *Progress in Mathematics*, pages 163–176. Birkhäuser Boston, 675 Mass. Ave., Cambridge MA, 1993.

[61] A. K. Lenstra and H.W. Lenstra, editors. *The Development of the Number Field Sieve.* LNM. Springer-Verlag, London, UK, 1st edition, 1993.

[62] A. K. Lenstra, H. W. Lenstra, Jr., M. S. Manasse, and J. M. Pollard. The Factorization of the Ninth Fermat Number. *Math. Comp.*, 61:319–349, 1993.

[63] A. K. Lenstra, H. W. Lenstra, Jr., M. S. Manasse, and J. M. Pollard. *The Number Field Sieve*, pages 11–41. In Lenstra and Lenstra [61], 1st edition, 1993.

[64] A. K. Lenstra and M. S. Manasse. Factoring with Two Large Primes. *Math. Comp.*, 63:785–798, 1994.

[65] P. Leyland, D. Leclair, R. Wackerbarth, and J. Gilchrist. NFSNET: Large-scale distributed factoring. `http://www.nfsnet.org/announcements.html`. (Aug. 2006).

[66] G. Malle. On the Distribution of Galois Groups. *Journal of Number Theory*, 92:315–329, 2002.

[67] G. Marsaglia, A. Zaman, and J. C. W. Marsaglia. Numerical Solution of some Classical Differential-Difference Equations. *Mathematics of Computation*, 53:191–201, 1989.

[68] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography.* CRC press, 1997.

[69] H. Mishima. World Integer Factorisation Center. `http://www.asahi-net.or.jp/~KC2H-MSM/mathland/matha1/` (Aug. 2006).

[70] C. Monico. GGNFS — A Number Field Sieve Implementation. Available at `http://www.math.ttu.edu/~cmonico/software/ggnfs/` (Aug. 2006).

[71] P. Montgomery. A Block Lanzcos Algorithm for finding Dependencies over $GF(2)$. In *Proc. EuroCrypt '95*, volume 921 of *LNCS*, pages 106–120. Springer-Verlag, 1995.

[72] P. Montgomery, S. Cavallar, and H. te Riele. A New World Record for the Special Number Field Sieve Factoring Method. Technical report, CWI, Amsterdam, 1997.

[73] P. L. Montgomery. Square Roots of Products of Algebraic Numbers, draft of 1997. Available at `http://ftp.cwi.nl/pmontgom/sqrt.ps.gz` (Aug. 2006).

[74] M. A. Morrison and J. Brillhart. A Method of Factoring and the Factorisation of F_7. *Math. Comp.*, 29:183–205, 1975.

[75] B. Murphy. Modelling the Yield of Number Field Sieve Polynomials. In *Algorithmic Number Theory, ANTSIII*, volume 1423 of *LNCS*, pages 137–151. Springer-Verlag, 1998.

[76] B. Murphy. *Polynomial Selection for the Number Field Sieve Integer Factorisation Algorithm.* PhD thesis, The Australian National University, 1999. Available at `http://web.comlab.ox.ac.uk/oucl/work/richard.brent/ftp/Murphy-thesis.ps.gz` (Aug. 2006).

[77] B. Murphy and R. Brent. On Quadratic Polynomials for the Number Field Sieve. In *Computing theory '98 (Perth)*, volume 20.3 of *Aust. Comput. Sci. Commun.*, pages 199–213. Springer, Singapore, 1998.

[78] P. Nguyen. A Montgomery-like Square Root for the Number Field Sieve. In *Algorithmic Number Theory, ANTS III*, volume 1443 of *LNCS*, pages 151–168. Springer-Verlag, 1998.

[79] O. Penninga. Finding Column Dependencies in Sparse Matrices over $\mathbb{F}_2$ by Block Wiedemann. Master's thesis, Leiden University, 1998.

[80] J. Pollard. *Factoring with Cubic Integers*, pages 4–10. In Lenstra and Lenstra [61], 1st edition, 1993.

[81] J. Pollard. *The Lattice Sieve*, pages 43–49. In Lenstra and Lenstra [61], 1st edition, 1993.

[82] C. Pomerance. The Quadratic Sieve Factoring Algorithm. In *Advances in cryptology (Paris, 1984)*, volume 209 of *Lecture Notes in Comput. Sci.*, pages 169–182. Springer, Berlin, 1985.

[83] C. Pomerance. A Tale of Two Sieves. *Notices Amer. Math. Soc.*, 43:1473–1485, 1996.

[84] R. L. Rivest, A. Shamir, and L. M. Adleman. A Method for obtaining Signatures and Public-key Cryptosystems. *Communications of the ACM*, 21:120–126, 1978.

[85] R. D. Silverman. Optimal Parameterization of SNFS. Available at `http://citeseer.ist.psu.edu/silverman03optimal.html`. (Aug. 2006).

[86] R. D. Silverman. The Multiple Polynomial Quadratic Sieve. *Mathematics of Computation*, 48:329–339, 1987.

[87] N. J. A. Sloane. Sequence A008290 in "The On-Line Encyclopedia of Integer Sequences.". `http://www.research.att.com/~njas/sequences/A008290` (Aug. 2006).

[88] I. N. Stewart and D. O. Tall. *Algebraic Number Theory and Fermat's Last Theorem.* Chapman and Hall Mathematics Series. Chapman and Hall, New York, USA, 3rd edition, 1987.

[89] E. Thomé. Fast Computation of Linear Generators for Matrix Sequences and application to the Block Wiedemann Algorithm. In *Proc. ISSAC 2001*, pages 323–331. ACM Press, 2001.

[90] S. S. Wagstaff, Jr. The Cunningham Project. `http://www.cerias.purdue.edu/homes/ssw/cun/` (Aug. 2006).

[91] C. Zhang. k-Semismooth Integers. Available at `http://www.czhang.net/research.html`. (Aug. 2006) pre-print, submitted to Math. Comp.